

PCTWORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06T 7/00	A2	(11) International Publication Number: WO 00/67204 (43) International Publication Date: 9 November 2000 (09.11.00)
<p>(21) International Application Number: PCT/US00/11919</p> <p>(22) International Filing Date: 2 May 2000 (02.05.00)</p> <p>(30) Priority Data: 60/132,324 3 May 1999 (03.05.99) US</p> <p>(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application US 60/132,324 (CIP) Filed on 3 May 1999 (03.05.99)</p> <p>(71) Applicant (for all designated States except US): PICTUALITY, INC. [US/US]; 139 Vista Del Grande, San Carlos, CA 94070-2930 (US).</p> <p>(72) Inventors; and (75) Inventors/Applicants (for US only): PAPA ZIAN, George [US/US]; 139 Vista Del Grande, San Carlos, CA 94070-2930 (US). DROPSHO, Steven [US/US]; 98 Spring Street, Apt.1, Amherst, MA 01002 (US).</p> <p>(74) Agent: WARHOLA, Stephen; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).</p>		<p>(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>Without international search report and to be republished upon receipt of that report.</i></p>
<p>(54) Title: IMAGE ANALYSIS PROCESS</p> <p>(57) Abstract</p> <p>A method of detecting target data in image data. The method comprises scanning elements contained in a data structure for image data; scoring the scanned image data, wherein a score value associated with the scanned image data indicates the likelihood of target data in the scanned image data; and viewing the scored image data for the target data.</p>		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

IMAGE ANALYSIS PROCESS

FIELD OF THE INVENTION

5 The present invention relates to an image analysis tool. More specifically, the present invention relates to an image analysis tool for analyzing and classifying images according to a image analysis protocol to determine if the image is pornographic in nature.

10 BACKGROUND

Image recognition is used to identify and classify images in accordance with an image recognition protocol. As such image recognition allows for the comparison of a reference image against another image or multiple images in
15 order to determine a "match" or correlation between the respective images. For instance, face recognition is based on matching a known facial model to a facial input image and determining a match between the respective images. In general, the input image must match some fairly strict rules so the system can quickly find the points for comparison. This is the basic technique employed in almost
20 all image recognition and classification database retrieval systems. As such, a variety of different image matching techniques have been employed to determine a match or correlation between images.

One such image matching technique is known as object classification. The object classification technique operates by segmenting the original image
25 into a series of discrete objects which are then measured using a variety of shape measurement identifications, such as shape dimensions and statistics, to identify each discrete object. Accordingly, each of the discrete objects are then classified into different categories by comparing the shape measurement identifications associated with each of the discrete objects against known shape
30 measurement identifications of known reference objects. As such, the shape measurement identifications associated with each of the discrete objects are

compared against known shape measurement identifications of known reference objects in order to determine a correlation or match between the images.

Another image matching technique utilized in determining a match between images is a process known as match filtering. Match filtering utilizes a pixel-by-pixel or image mask comparison of an area of interest associated with the proffered image against a corresponding interest area contained in the reference image. Accordingly, provided the area of interest associated with the proffered image matches the corresponding interest area of the reference image, via comparison, an area or pixel match between the images is accomplished and the images are considered to match.

Yet another technique utilizes a series of textual descriptors which are associated with different reference images. The textual descriptors describe the image with textual descriptions, such as shape (e.g., round), color (e.g., green), and item (e.g., ball). Accordingly, when a proffered image is received from comparison, the textual descriptor of the proffered image is compared against the textual descriptors associated with the reference images. As such, the textual descriptor associated with the respective images under comparison are compared to each other in order to determine a best match between the textual descriptions associated with each image, and therefore, a match between the respective images.

Each of the aforementioned image matching techniques utilize different types of data or partial image data to describe the images under comparison, however, these techniques are generally not reliable in identifying pornographic images or images containing pornographic content. The image metrics of pornographic images do not render themselves to easy classification. For instance, matching methods involving shape have been tested, however, shape is not a reliable metric for pornographic images since the subjects can be in any position or the camera can be at different distances from the subjects. Research groups have tried to match limbs, which is extremely hard to do, and the performance has been unsatisfactory.

In one instance, in order to overcome variation in camera position a general technique has been developed that uses Gaussian filters based on second or higher order derivatives of certain qualities of the image. These derivatives are invariant to simple rotation and translation in the image plane and somewhat
5 robust to rotation orthogonal to the image plane. While an interesting technique, the generality of this method limits the recall and precision of proper image analysis and classification.

As a result, there is a need for an image recognition or detection technique that is reliable in identifying pornographic images or images
10 containing pornographic content.

SUMMARY OF THE INVENTION

A method of detecting target data in image data. The method comprises
15 scanning elements contained in a data structure for image data; scoring the scanned image data, wherein a score value associated with the scanned image data indicates the likelihood of target data in the scanned image data; and viewing the scored image data for the target data.

BRIEF DESCRIPTION OF THE DRAWINGS

20

The present invention is illustrated by way of example in the following drawings in which like references indicate similar elements. The following drawings disclose various embodiments of the present invention for purposes of
25 illustration only and are not intended to limit the scope of the invention.

Figure 1 illustrates an embodiment of a suitable computing environment in which the present invention may be implemented in accordance with the teachings of one embodiment of the present invention.

Figure 2 illustrates an embodiment of an exemplary network
30 environment in which the present invention may be employed in accordance with the teachings of one embodiment of the present invention.

Figure 3 illustrates an embodiment of a high-level general overview of the image analysis / detection process used in determining whether an image contains pornographic content in accordance with the teachings of one embodiment of the present invention.

5 Figure 4 illustrates an embodiment of a scanning process that may be implemented within the present invention in accordance with the teachings of one embodiment of the present invention.

Figure 5 illustrates an embodiment of a scoring process that may be implemented within the present invention in accordance with the teachings of
10 one embodiment of the present invention.

Figure 6 illustrates an embodiment of a viewing process that may be implemented within the present invention in accordance with the teachings of one embodiment of the present invention.

Figure 7 illustrates an embodiment of a archiving process that may be
15 implemented within the present invention in accordance with the teachings of one embodiment of the present invention.

Figure 8 illustrates an embodiment of a typical video / movie data segment to which one embodiment of the present invention may be applied in accordance with the teachings of one embodiment of the present invention.

20 Figure 9 illustrates an embodiment of a Broadcast Market to which one embodiment of the present invention may be applied in accordance with the teachings of one embodiment of the present invention.

Figure 10 illustrates an embodiment of a system capable of implementing the teachings of the present invention within a V-Chip
25 environment in accordance with the teachings of one embodiment of the present invention.

Figure 11 illustrates an embodiment of a machine-readable medium capable of implementing the teachings of the image analysis / detection process of the present invention in accordance with the teachings of one embodiment of
30 the present invention.

DETAILED DESCRIPTION

The following detailed description sets forth numerous specific details to provide a thorough understanding of the invention. However, those of ordinary skill in the art will appreciate that the invention may be practiced without these specific details. In other instances, well-known methods, procedures, protocols, components, algorithms, and circuits have not been described in detail so as not to obscure the invention.

In one embodiment, the steps of the present invention are embodied in machine-executable instructions, such as computer instructions. The instructions can be used to cause a general-purpose or special-purpose processor that is programmed with the instructions to perform the steps of the present invention. Alternatively, the steps of the present invention might be performed by specific hardware components that contain hardwired logic for performing the steps, or by any combination of programmed computer components and custom hardware components.

Computer Environment

Figure 1 and the following description are intended to provide a general description of a suitable computing environment in which the invention may be implemented. Although not necessarily required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a computer, such as a client workstation or a server. Generally, program modules include routines, programs, objects, components, data structures and the like that perform particular tasks or implement particular abstract data types.

Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers and the like. The invention may also be practiced in distributed computing

environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

5 As shown in Figure 1, an exemplary general purpose computing system may include a conventional personal computer 20 or the like, including a processing unit 21, a system memory 22, and a system bus 23 that couples various system components including the system memory 22 to the processing unit 21. The system bus 23 may be any of several types of bus structures
10 including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory 22 may include read-only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system 26 (BIOS), containing the basic routines that help to transfer information between elements within the personal computer 20, such as
15 during start-up, may be stored in ROM 24.

The personal computer 20 may further include a hard disk drive 27 for reading from and writing to a hard disk (not shown), a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a
20 CD-ROM or other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 may be connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical drive interface 34, respectively. The drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data
25 structures, program modules and other data for the personal computer 20.

Although the exemplary embodiment described herein may employ a hard disk, a removable magnetic disk 29, and a removable optical disk 31, or combination thereof, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible
30 by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read-only

memories (ROMs) and the like may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37 and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite disk, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus 23, but may be connected by other interfaces, such as a parallel port, game port, or universal serial bus (USB). A monitor 47 or other type of display device may also be connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor 47, personal computers may typically include other peripheral output devices (not shown), such as speakers and printers.

The personal computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. The remote computer 49 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 20, although only a memory storage device 50 has been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area network (LAN) 51 and a wide area network (WAN) 52. Such networking environments are commonplace in offices, enterprise-wide computer networks, Intranets, and the Internet.

When used in a LAN networking environment, the personal computer 20 is connected to the LAN 51 through a network interface or adapter 53. When used in a WAN networking environment, the personal computer 20 typically includes a modem 54 or other means for establishing communications over the wide area network 52, such as the Internet. The modem 54, which may be

internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections
5 shown are exemplary and other means of establishing a communications link between the computers may be used.

Network Environment

As noted, the general-purpose computer described above can be
10 deployed as part of a computer network. In general, the above description applies to both server computers and client computers deployed in a network environment. Figure 2 illustrates one such exemplary network environment in which the present invention may be employed. As shown in Figure 2, a number of servers 10a, 10b, etc., are interconnected via a communications network 160
15 (which may be a LAN, WAN, Intranet or the Internet) with a number of client computers 20a, 20b, 20c, etc. In a network environment in which the communications network 160 is, e.g., the Internet, the servers 10 can be Web servers with which the clients 20 communicate via any of a number of known protocols such as, for instance, hypertext transfer protocol (HTTP). Each client
20 computer 20 can be equipped with a browser 180 to gain access to the servers 10, and client application software 185. As shown in the embodiment of Figure 2, server 10a includes or is coupled to a dynamic database 12.

As shown, the database 12 may include database fields 12a, which contain information about items stored in the database 12. For instance, the
25 database fields 12a can be structured in the database in a variety of ways. The fields 12a could be structured using linked lists, multi-dimensional data arrays, hash tables, or the like. This is generally a design choice based on ease of implementation, amount of free memory, the characteristics of the data to be stored, whether the database is likely to be written to frequently or instead is
30 likely to be mostly read from, and the like. A generic field 12a is depicted on the left side. As shown, a field generally has sub-fields that contain various

types of information associated with the field, such as an ID or header sub-field, type of item sub-field, sub-fields containing characteristics, and so on. These database fields 12a are shown for illustrative purposes only, and as mentioned, the particular implementation of data storage in a database can vary widely according to preference.

Thus, the present invention can be utilized in a computer network environment having client computers for accessing and interacting with the network and a server computer for interacting with client computers and communicating with a database with stored inventory fields. Likewise, the image analysis process of the present invention can be implemented with a variety of network-based architectures, and thus should not be limited to the examples shown. The present invention will now be described in more detail with reference to preferred embodiments.

15 Image Analysis

The present invention is directed to a detection method or process, also referred to as an image analysis system (system) or core analysis program, wherein an image is analyzed to determine if the image contains pornographic content. Figure 3 illustrates a high-level general overview of the detection process used by the present invention in determining whether an image contains pornographic content. As shown in Figure 3, the general detection process (main loop process) comprises a scanning process, scoring process, viewing process, and archiving process. It is understood that different process segments (e.g., scanning process, scoring process, viewing process, and archiving process) may be used independently of the main loop process, as required or as desired for a particular user implementation.

The scanning process allows a user to designate different data locations (e.g., directories or files) to be analyzed in order to determine if the data locations contain pornographic image content. The scoring process is used to score or assign a ranking to the images, wherein the score value indicates the likelihood of whether the scanned image contains pornographic content or not.

The viewing process allows a user or administrator the option to view the scored images to determine whether or not the scored images are pornographic in nature, as opposed to non-pornographic in nature, before addressing the matter.

The archiving process allows the user the ability to select images and save
5 images for an administrative report on such activity.

The detection process may be implemented in a variety of different ways, such as but not limited to, as an application sitting on a fixed hard drive, CD drive, or other non-volatile media, in addition to the implementation described above. The detection process may be implemented as a continuously
10 running daemon that is activated via network requests and returns results to the remote initiator. In one embodiment, the detection process may be configured to operate from a CD ROM and as such does not require local storage, or the detection process can be installed on the main hard drive as any other application, or loaded from ROM, FlashMemory or other non-volatile memory
15 storage and run in embedded environments such as a TV set, VCR, or any autonomous computing environment.

In one embodiment, the present invention uses a GUI interface with common drop down menus to give users an easy and familiar interface to implement or operate the detection process. The menu items give users the
20 ability to set paths, view images, delete files, print reports, direct output, and the like. Likewise, the detection process allows the user to first select the directory path that is to be searched. The interface displays a common file and folder directory structure and the user selects a directory to be searched. The detection process then examines each file in the selected directory and any corresponding
25 subdirectories. The detection process then goes through the entire selected section and finds each image file of a format recognized by the detection process. The detection process can recognize and score files in all the generally known common formats. Additional formats can be added by simply adding a module for identifying the format and decoding the format.

30 Accordingly, the detection process reads the contents of each file to identify image types by their respective content and not by the easily falsified

extensions as is generally done by other image reading software. The detection process records selected file types, for which it does not contain decoders, and reports these files in a concise manner to simplify and speed manual processing, if such is desired.

5 Generally, the detection process can be extended to recognize all image files in any format. For instance, the detection process may be configured to find images in MS Word documents, spreadsheets, and e-mail attachments. In one embodiment, the detection process finds any images rapidly via a disk scan and parsing of the beginning portion of the file. In one embodiment, the image
10 format identification and decoding code used may be obtained from ImageMagick™ software. Typically, the process of reading and decoding code is not integrated into the core analysis program, as such, the code can easily be replaced by any other decoding library, if so desired. The scanning sifts through gigabytes of data for the user in minutes. Once all the files are found, the
15 detection process then scores these images for the likelihood of their being pornographic. The user has the option to have the images automatically displayed as they are scored to give the user a preview of the types of images on the disk before scoring has completed. The user then is prompted to start the Viewer. Additionally, the user may save the results of the detection process, via
20 the archiving process, which allows the user the ability to select images and save images for an administrative report on such activity, if such is desired.

Figure 4 illustrates one embodiment of a scanning process that may be implemented within the present invention. Initially, at block 400, as illustrated in Figure 4, the user may specify the location of data structures (e.g., directories,
25 files, etc.) which are to be scanned, whereupon the designated data structures are scanned to determine whether the specified data structures contain image files or image data. The terms image data, image files, and images are used to denote data containing graphic images or image data, such as but not limited to, visual images, video image content, and the like. In one embodiment, the
30 scanning process may be configured to identify whether or not the data structure contains images using a variety of different techniques, such as but not limited

to, examining the extension identifiers (e.g., GIF, JPEG, VIDEO, etc.) associated with the data to determine whether the data in the data structures contain image files or image data.

In one embodiment, the scanning process can be performed through standard system calls that update the access dates associated with the data or can be performed via an explicit read of the disk sectors bypassing the standard system calls which maintains the access date in its original state. This is essential in tasks where one does not want to compromise the integrity of the disk being scanned, e.g., in evidence for law enforcement or a customs official checking a system.

Generally, the directory structure of any modern file-system is a tree-like structure with a top-most directory called the root directory. Generally, within a directory, files and sub-directories are placed. Generally, a file is a set of data the file system considers as a unit, which does not typically contain other files or directories (other applications may interpret the file data differently, of course). Directories contain information about the files they contain and may contain other directories. The file-system provides functions to list contents in a directory, read file information, and read file data. Using these basic functions most directories and files can be discovered from a given starting directory.

Further, Unix file-systems contain the concept of a link. A link is generally a pointer to another file or directory that exists somewhere else in the file-system. Actions on the link redirect the file-system to the where the actual file resides. With links, the normally tree-like file-system can point a sub-directory back to the root directory to create a cycle. If this cycle is not detected then traversing the directory structure will loop back on itself and never complete. Fortunately, links are easily detected and are not traversed, thus breaking any possible cycles and ensuring the traverse of the directory will visit each file and sub-directory once only.

Generally, there are two commonly known methods of traversing a tree-like structure (e.g., data structure): breadth first search (BFS) and depth first search (DFS). In a preferred embodiment, the present invention uses the DFS

method and its associated algorithm as detailed in the Scanning Process Flowchart of Figure 4. The DFS method processes files found first, before directories, and processes each directory completely (e.g., all files and subdirectories) before the next directory.

5 In an alternate embodiment, the BFS method may be employed to traverse a data structure. The BFS method, however, also processes the files before directories, but then reads each directory and adds the contents to the flist and dlist before processing the sub-directories. While BFS has some advantages in certain applications, those advantages are not necessarily required
10 here. The advantage DFS enjoys over BFS in this application is less space requirements by keeping fewer items on the flist and dlist structures.

 In one embodiment, the present invention implements the depth first search (DFS) as follows. The contents of a directory are read with files put into a file list (flist) and directories put onto a directory list (dlist). In one
15 embodiment, to keep the lists as small as possible, all files on the flist are processed immediately to detect any images whose paths are recorded in an image database for later detailed processing. When the flist is empty then the next directory in the dlist is selected and its contents read, repeating the steps to process files and directories found. The search ends when both the flist and
20 dlist are empty.

 In one embodiment, image files are detected by opening the file and comparing select bytes for expected values of each of the image format signatures. If a match is found then the file is stored in the image database along with the image type to which it matched. Reading the file can have side effects
25 in some file-systems. Specifically, in the Windows/NT system file access dates are modified to reflect this read. While the date information can be read and stored before any modification to these dates occurs, in some environments it is preferred that the dates do not change.

 To accomplish this, low-level file-system reads can be performed that
30 bypass the operating system's file manager. Such calls read the disk directly to read raw sector data which must be interpreted to delineate directories and

recreate files. The advantage of bypassing the operating system file manager is to the eliminate side effects that change file state. In Windows/NT systems these calls are to the BIOS via a set of well-known DOS application programmer's interface calls. As such, the disk scanning process of may be
5 implemented through standard system calls that update the access dates or can be performed via an explicit read of the disk sectors bypassing the standard system calls, which maintains the access date in its original state. This is essential in tasks where one does not want to compromise the integrity of the disk being scanned, e.g., in evidence for law enforcement or a customs official
10 checking a system.

Accordingly, referring back to Figure 4, the scanning process starts at Block 400 with a user selecting a desired directory (e.g., the start directory), the scanning process then proceeds to traverse all files and directories that reside under the start directory. The start directory can be any directory that is
15 accessible to the user or system. The medium can be floppy disk, hard drive, CD-ROM, DVD, Zip/Jaz/SuperFloppy disk, or any other medium upon which a supported file system can be written or otherwise stored. At Block 400, the data structures selected for the scan are initialized, whereupon the start path is verified to ensure the start path valid and the main scan loop is entered. In one
20 embodiment, this includes creating empty lists to store the directories (dlist) and files (flist) that are found and waiting to be processed. Additionally, any images previously stored in the image database may be cleared to remove any images from any previous scans.

At Block 405, the scan process checks if the file list (flist) is empty. If
25 the file list (flist) is empty, then the scan process will check the directory list. If file list (flist) is not empty, then the scan process will process the file.

Next, at Block 410, the scan process obtains elements from the end of the file list (flist), decreasing the number of elements in the file list (flist) by one, to test the element to determine if the element is or contains an image.

30 Accordingly, at Block 415, the scan process reads data, such as a number of bytes, from the image file that will contain the signature of any

image type supported. For each image type, a check is made to see if the proper signature values are present in the proper byte positions. If not, then the file is not classified as an image and all data structures allocated for its information are reclaimed and the process branches to Block 405. If the file is an image, then it
5 is forwarded to Block 420 for processing.

At Block 420, the file, which has been identified as an image, is added to a storage medium, such as to an image database, for subsequent processing. The information associated with the file which is recorded or stored may include, but is not limited to, the creation, modified, access dates, the images
10 size (the number of pixels, not file size), the full path to the image, and the image type. Accordingly, upon recording the file in the image database, the process branches to Block 405.

As mentioned above, at Block 405, the scan process checks if the file list (flist) is empty. If the file list (flist) is empty, then the scan process will check
15 the directory list (dlist). Accordingly, at Block 425, the process checks the directory list (dlist) to determine if the directory list (dlist) is empty. Provided the directory list (dlist) is empty, the scanning process is completed and scanning process branches to Block 400 or Block 405. Alternately, provided the directory list (dlist) is not empty, the scanning process proceeds to Block
20 430 to process the element contained in the directory list (dlist).

Accordingly, at Block 430, the scanning process obtains elements from the end of the directory list (dlist), decreasing the number of elements in the directory list (dlist) by one, to process the selected element.

Next at Block 435, the scanning process performs a read of the current
25 directory (CurDir) to obtain a list of directories and files that are to be put on the proper list for processing.

At Block 440, the scanning process then takes the last element contained in the current directory (CurDir) to check if the element is a file or directory.

At Block 445, the element is tested to see if the element is NULL.
30 Provided the element is NULL, then CurList is empty and all files and directories have been placed on the proper lists, whereupon the scanning process

branches to Block 405, to process any other file and directory lists. Provided the element is not NULL, then the element is examined to determine if the element is a file.

Accordingly, at Block 450, the element type is analyzed to ascertain
5 whether the element is a file or a directory element. The scanning process may use a file system call in order to assist in ascertaining whether the element type is a file or a directory element.

Provided the element is a file, then at Block 455, the element path (e.g., file path) is added onto the file list (flist), whereupon the scanning process is
10 branched back to Block 440 to process remaining elements.

Alternately, at Block 460, provided the element is a directory, then at Block 465, the element path (e.g. directory path) is added onto the directory list (dlist), whereupon the scanning process is branched back to Block 440 to process remaining elements.

15 At Block 470, provided the element is not a file nor a directory, then an error occurred or it is an entry of no interest. Accordingly, the element is skipped and the scanning process is branched back to Block 440 to process remaining elements.

Accordingly, provided the data structure contains images, the image file
20 information is designated to be analyzed by the scoring process. In one embodiment, the image file information (e.g., pathname, filestamps, etc.) is added to an image database for eventual analysis by the scoring process. In an alternate embodiment, the image file information is passed directly to the scoring process for real-time analysis and scoring of the image file or image
25 data.

Figure 5 illustrates an embodiment of a scoring process that can be used to score or assign a ranking to the images, wherein the score value indicates the likelihood that the scanned image contains pornographic content. In one embodiment, the scoring process sorts the images by the internal score assigned
30 during the scoring phase. Generally, a higher score denotes more elements of a pornographic image, and correspondingly, a lower score denotes less elements

of a pornographic image. In one embodiment, the images are sorted in descending order thus the images most likely to contain nudity or pornography appear first.

5 In one embodiment, the scoring process employs a set of filters tuned to select pixels in an image that have a high probability of representing skin on the arms, legs, chest, abdomen, etc. These regions are referred to as target data or regions of interest. Skin on the face, fingers, and toes is typically filtered out.

10 In one embodiment, the image filtering employed by scoring process is performed with reference to some, or all, of the following qualities or assumptions of the target data: (1) Skin consists of distinctive colors. (2) A typical patch of skin consists of a relatively narrow range of colors. (3) Within the range of skin colors, skin exhibits a complex texture from slight variations of color and intensity in neighboring pixels. (4) Interesting skin regions consist
15 of a relatively large set of pixels, thus, skin pixels should generally adjoin other skin pixels. (5) Colors occur with a limit on their frequency within a patch of skin. Too much of a color indicates a non-skin region. An estimation of the population distribution of color occurrence frequencies can be learned from sample skin patches. (6) Too much or too little color variation indicates regions
20 that the eye perceives as too smooth or too coarse to be skin. (7) Skin is highly reflective. Some regions appear white, exhibiting little texture and are difficult to differentiate from common, non-skin flat color regions. (8) Dark regions, possibly due to shadow, exhibit a shift towards the blue spectrum, and, thus, are unreliable indicators of true color. (9) Images of biological entities invariably
25 create curved lines or edges in the image. Man-made structures typically exhibit angular edges that lack curves. Lack of curved edges indicates an absence of people and, hence, target content. (10) Poses of people in pornographic images are unconstrained. Often images show the back of the head, no head, side views of a face, partial display of limbs, or overlapping limbs from multiple people.
30 Thus, techniques that rely on detecting faces or limbs or assume the subjects are in a limited range of poses are fundamentally restricted in their accuracy due to

these assumptions. (11) In pornographic images, the chest and genitalia- the typical components of interest are statistically most commonly pictured in the lower 60% of the image and generally centered. This fact permits processing to be focussed in the region most likely to contain pornographic imagery. (12)

5 Pornographic images have only a small number (1 to 5 regions) of distinct regions of skin. Thus, a highly fractured image having many more regions labeled as skin is unlikely to actually represent skin. (13) Some forms of digital image representation have a limited number of permissible colors that can be supported in one image. Both the GIF and TIFF formats restrict the color
10 palette to only 256 colors. In contrast, the JPEG format supports 65,000+ to 16 million colors. Thus, GIF and TIFF images exhibit poor skin texture, and typically have large regions of a single color. In response, a compensation technique to recover the skin texture from such images may be implemented, discussed in further detail below. (14) The more interesting pornographic
15 images tend to have relatively good detail so they tend to be large in comparison to most non-target images where detail is typically compromised for compactness. (15) Target images may exhibit strong symmetry along the vertical axis (left vs. right halves) but usually little to no symmetry along the horizontal axis (top vs. bottom) or along either diagonal. (16) Images are of
20 differing scales. Small images, e.g., thumbnails, lack detail that large images typically have. Thus, a good analysis technique must consider image size and the appropriateness of filters for that size.

Color is a primary filter in the analysis process, however, color alone gives limited accuracy in reliably detecting pornographic content. Additional
25 filters include texture analysis that pick up the unique texture qualities in skin, color compensation for images using inferior encoding formats, curvature detection, and symmetry analysis.

Referring to Figure 5, at Block 500, the scoring process is initialized, wherein any previous scores associated with images may be cleared from the
30 system, if such is desired. In one embodiment, a user specifiable filter that is dynamically updated by the filtering process to indicate the pixels that are to be

omitted or otherwise not considered during the scoring process may be implemented into the scoring process. Accordingly, in one embodiment, the filter that tracks or specifies which pixels are to be omitted may be cleared, thereby allowing all pixels to be initially to be accepted. Additionally, in a preferred embodiment, any edge markings associated with the edge structure of an image are removed.

Next at Block 505, the scoring process determine whether there are any current images that are to be analyzed and scored. Accordingly, the scoring process checks to find the next image to be scored (e.g., images awaiting the scoring process). Provided unscored images exist, then the scoring process proceeds to Block 510, wherein the scoring process reads the next unscored image and decodes the data into a representation used in the scanning process. Alternately, provided there are no unscored images existing the scoring process exits.

At Block 515, the scoring process sub-samples the selected image, wherein a subset of the pixels, generally evenly distributed throughout an image, e.g., every Nth pixel, is selected. In one embodiment, to speed processing, a sub-region of the image is used, as such the image analyzed may be limited to an image comprising no more than MAXSIZE pixels (in an Width x Height ratio). Further, the image can be reduced or shrunk either by properly averaging surrounding NxM pixels to maintain full image integrity, or more quickly by simply taking every Nth pixel in every Mth row using a fixed stride, or use a random selection process within the NxM rectangle. In a preferred embodiment, the scoring process uses a fixed stride for maximum speed.

From general observation, target images tend to have the target data clustered in the lower half of the image and generally centered, though web advertisements are less consistent in their positioning of target data. Accordingly, in one embodiment, the sub-samples are configured, without smoothing, from the bottom portion of the image so that it contains no more than 16,384 total pixels. Configuring the sub-sampled image to this particular number of pixels has been chosen to match well with the processing speed of

current technology. Scaling the image to a fixed size sets an upper bound on processing time. In one embodiment, images that are already below the size threshold are processed in their entirety.

Accordingly, at Block 520, the scoring process applies a color
5 compensation scheme to the selected image (e.g., sub-sampled image). Generally, images encoded in formats with a limited color range lose their color complexity due to mapping the full range of color into a much smaller set of colors, e.g., a palette of only 256K. Typically, this simplification is done to decrease the image size, but much of the subtle texture information may be lost.
10 The result of using such a technique is to over represent colors and generate swatches of constant color. The effect makes skin appear artificial or computer generated.

Accordingly, a method of retrieving the lost texture in skin, or any other non-computer generated image, is to re-encode the image in a format that
15 supports a wide range of colors and is designed to smooth transitions in color. The JPEG format does precisely this. JPEG was designed for higher resolution images and supports at least 64K colors. JPEG detects changes in colors and tries to make a smooth transition between values by adding intermediate colors. Unfortunately, while an actual conversion to a color rich encoding, such as
20 JPEG, is the proper compensation method, it is relatively time consuming in today's processing technology.

Accordingly, in one embodiment, the scoring process approximates this process by smoothing the image, i.e., by averaging the values surrounding a pixel. Accordingly, the effect on images that are actually computer generated
25 and meant to have regions of constant color is to blur the edges between color boundaries. Fortunately, in these types of images the constant color swatches are much broader than in non-computer generated images so that the increase in texture is generally limited to the boundaries and of minimal area in most cases.

At Block 525, the scoring process marks pixels in the selected image
30 which are surrounded by identically colored pixels. Accordingly, the marked pixels are effectively removed or marked as omitted from the image. As is

understood, skin is generally not a flat constant color; therefore by omitting pixels that are surrounded by identically colored pixels items such as computer generated borders, graphics, and the like are effectively omitted from the image. For instance, in one pass the scoring process may filter or mark as omitted those
5 pixels from an image having neighbors that are of identical color.

Next at Block 530, the scoring process marks pixels in the selected image, which are not capable of being mapped, to a designated accepted color set defined by the color map. In one embodiment, as skin contains a wide range of colors, the scoring process employs the use of a color histogram trained on
10 target data to define the set of acceptable colors. In one embodiment, to define the set of acceptable colors, a set of known target data is streamed through a training filter that treats the RGB color as a vector in a three dimensional space using the red, green, and blue component values, respectively.

In one embodiment, the 3D space is of dimension $255 \times 255 \times 255$ that can
15 be compressed into an $N \times N$ 2D space. Generally, the first index into the 2D space is formed by scaling to the range $[0, N]$ the angle between an axis and the 2D vector formed by two of the color components, e.g. RG. The second index is formed by scaling the angle between the 2D vector and the RGB 3D vector. This is just one method to create a color mapping insensitive to intensity. In a
20 preferred embodiment, a value of 128 is used as the value for N.

In one embodiment, at each color in the mapping, a fraction of how frequently that color typically should appear in target data is generated, which is a space efficient approximation of a population distribution for each color and is obtained from training on known samples of skin imagery.

25 In one embodiment, during the image scanning process, a mask is created for the image of which pixels are of acceptable color and a summation of the number of occurrences for each color. Determining the color of low intensity regions (appearing black) is unreliable, but the errors tend to move the color towards the blue spectrum so this error adds few false positives. To
30 further reduce the effects of this shift, a threshold intensity can be set below which a pixel is deemed too unreliable to be considered. Similarly, at high

intensity values, color is again unreliable due to saturation in the optical sensor or image capturing media. At high intensity (near white) regions appear flat in texture. Again, a threshold can be used above which a pixel is considered too saturated to provide reliable color. In one embodiment, the threshold values are
5 determined empirically.

The resulting mask, after the color match, has each pixel set as pass or not pass. At this point an erode/dilate phase is performed, which removes pixels classified as passing but which have insufficient support from neighboring pixels to justify that classification. In one embodiment, for each
10 pixel that did not pass, the pixel's immediate eight neighbors are set as also not passing. This process is referred to as croding the 'pass' regions in the mask. Since skin regions should be of some significant size, this action removes individual pixels that do not have sufficient support from the neighboring pixels. On a second pass, at each pixel that still is marked as 'passed' the eight
15 nearest neighbors are marked as also having passed. Opposite to erosion, this action dilates the 'pass' regions. What regions remain have strong local color evidence for skin. As is understood, any of the commonly known texture classification methods may be used on the mask rather than the dilate/erode process. The terms grow/shrink are used interchangeably with dilate/erode.

20 At Block 535, the scoring process analyzes the image to compare the image to a specified image size (c.g., user or system specified MINSIZE), if the image is smaller than a minimum size (MINSIZE) then scoring process proceeds to Block 545, bypassing image texture analysis. Alternately, provided the image is larger than the minimum size (MINSIZE) then scoring process
25 proceeds to Block 540, to perform image texture analysis.

Accordingly, at Block 540, an image texture analysis is performed on the image, wherein select pixels are marked as omitted that are in regions estimated to be too rough to represent skin. As is understood, texture is a pattern of variation over a region. The variation can be of color, intensity, or
30 some other measure. In one embodiment, the texture analysis finds edges via the difference of intensities between pixels along both the horizontal and

vertical directions in the various color planes. Edges are points where variations in texture occur. In one embodiment, a simple convolution with a 3x3 averaging edge detection mask, such as the Sobel operator, may be employed and is widely known in the image processing community. In an alternate
5 embodiment, other edge detectors are more robust but require considerably more processing resources.

One embodiment of a 3x3 averaging edge detection mask is illustrated as follows. Detection of a horizontal edge uses the following 3x3 mask:

```
1 1 1
10 0 0 0
1 1 1
```

Detection of a vertical edge uses the following 3x3 mask:

```
1 0 1
1 0 1
15 1 0 1
```

The smooth nature of skin implies variations are small, so a threshold value (determined from the training data) suppresses edges with small differences. The edges from the four steps are coalesced into a single mask. The mask now represents the edges of the image and the binary settings can be
20 interpreted as a texture of edges for the image. Any method of texture analysis can be used here. For efficiency purposes, a method that is fast and greedy may be implemented. Since highly textured regions will have many edges in the edge mask, growing the edge regions (a process identical as that done in the mask after color filtering) creates solid areas in the edge mask that remain after
25 shrinking.

One possible problem with edge detection is that a fixed threshold does not scale well with intensity. A 5% threshold tolerance suppresses edges for variations up to a difference of 10 in contrast at an intensity of 200, but places an edge at any intensity change at low intensities of 20. Thus, shadows are
30 marked as highly textured when in fact the eye perceives the region as smooth. Compensation can be attained by performing the textured region shrink phase

multiple times after the grow phase to compensate for area lost to shadowing in skin regions. In a preferred embodiment, performing the textured region shrink phase at least 4 times has empirically yielded the appropriate compensation.

In an alternate embodiment, texture via zero crossings of the second
5 derivative of the intensity values may be used as a compensation technique. Similarly, this is generally done in both the horizontal and vertical directions and in the various color. A single pass in one direction is given for purposes of the following example, only.

Initially, an image is created with the difference values between each
10 pair of pixels by way of a 3x3 convolution using an edge detector mask, such as the Sobel operator. Along the chosen direction, mark in the zero crossing mask points where the difference values change from positive to negative values, or vice versa. This basic operation is a second derivative in the horizontal direction. A simple Laplacian mask could be used, however, the averaging
15 done by the edge detection operator helps significantly. Shadow regions on the curved portion of the body will have a monotonic shift in intensity values until near the edge of the body part. Zero crossing detection will detect the edge of the body part nearly independent of the shadowing. In contrast, highly textured regions will have frequent zero crossings as the intensity values shift from light
20 to dark and back again across a small area.

Generally, the mask of zero crossings has more precise information but is more sparse than that for edge detection. As such, the process of growing and shrinking to form the textured regions generally does not work well. Instead, the density of zero crossings per unit area (e.g., 7x7 region) is calculated around
25 the pixel of interest and use a simple threshold to classify the pixel as target data or non-target data. The threshold value is determined from the training set of target data and the training set of highly textured non-target data. The result is a mask marking target and non-target regions. Accordingly, the grow-shrink process is applied on the target data to remove incidental pixels that are marked
30 opposite the categorization supported by the neighboring pixels. The previous description of approximate texture classification is used for its efficiency,

simplicity, and speed. Any more traditional texture classification method will work as well, but are generally more computationally intensive and slower. In an environment where the low-end processing speed can be bounded, more aggressive and computationally intensive texture classification algorithms can be employed.

Texture classification has the additional benefit of suppressing the contribution of facial features, hands, and feet to the final score due to their apparent texture from shadowing between the fingers, toes, eyes, nose, ears, and hair. In one embodiment, a color-based texture classification scheme can also be employed. The color-based texture classification scheme is a combination of intensity texture classification methods on the individual color bands.

In one embodiment, Heuristics are added to omit the texture analysis phase on small images and on images where the color filter resulted in only a small region of acceptable color.

At Block 545, the scoring process merges the resulting structures, called maps, that represent which pixels are omitted from the previous filters. Merging is the process of marking a pixel (e.g., in Final) as omitted, if the corresponding pixel was omitted by FilterByColor or TextureAnalysis. As such, the scoring process merges the color and texture masks into a final mask. The final mask classifies each pixel as target data provided both masks have the pixel classified as target data.

At Block 550, the scanning process groups pixels into regions and removes those regions too small to be of interest.

Next at Block 555, the scanning process generates an initial score from the ratio of pixels not omitted in the Final Map. As such, initial score is the ratio of the number of remaining pixels (those not omitted) to the total pixel count (number remaining plus number omitted), or alternately expressed as the ratio of pixels kept relative to the total number of pixels analyzed.

At Block 560, the scoring process performs color compensation that depresses the contribution of pixels whose color is over represented relative to the expected contribution from the training map.

In one embodiment, this phase takes a global view of the image and performs an adjustment to the total count of target pixels using a rough approximation for color population distribution. The color classification filter provides a count of how many times each color occurred in the image. From the training data is an upper value of the fraction a color should appear in a target region. Call the sum of the remaining pixels in the image the active total. In one embodiment, for each color, calculate the fraction of the time it occurs in the active total. A color that occurs more often than expected is corrected downward. One that occurs less often is not adjusted. The correction takes the form of decreasing the number of occurrences by a factor equal to the ratio of the actual fraction to that of the expected. For example, a color that should occur 15% of the time or less but actually occurs 30% across the regions after filtering has its count corrected to only half of that allowed, or 7.5% ($15\%/30\% = 0.5$, $15\% \times 0.5 = 7.5\%$). This severely penalizes large areas that are unlikely to be target data, but that passed the previous filters. The greater the deviation from the expected data, the more severely its contribution to the score is decreased, while small deviations from the expected are only marginally effected. For a skin color map, tan walls are one example where a color can be greatly over represented.

Next at Block 565, the scoring process calculates a Color Count Factor, which is a value representing the perceived range of colors in an image. In one embodiment, this value is used to determine images of limited color.

Images having a limited range of color lack an essential quality that most pornographic images share, which is skin tone. Lacking this key quality, such images will score low on any color-based filter. Thus, the ability to quickly determine images which appear to people to be relatively monochrome is an important capability for providing the ability to group this inherently difficult class of images for fast viewing. The scoring process uses a technique to determine a relative count of effective colors in an image, upon which a threshold can be used to differentiate images of perceived limited color range.

The result of such a classification is to enable a viewer that displays only images of limited color for quick manual review.

Determining the exact number of colors in an image is a computationally expensive operation. The technique used by the scoring process is used for the purposes of determining a relative count of humanly perceived colors with the intent of establishing a threshold below which a person interprets the images as roughly monochrome.

Since the eye perceives colors that are close in the spectrum as the same color, the technique arbitrarily bins the full color range into a smaller range. The particular binning factor is a minor efficiency that is relative to the color map used, such as a 129x129 map, and may differ given a different color mapping. Any binning method that groups colors of nearly imperceptible visual difference together can be used. Regardless of the binning, the main goal is in determining the contributing factor of each color in the image. A straight count of the number of colors that occur implicitly weights the contribution of each color equally. People, however, perceive the contribution of a color relative to its frequency in an image. Thus, a unique feature of the scoring process is to base the contribution factor of a color relative to the fraction that it occurs in the image.

In one embodiment, the metric for the number of colors is a value that ranges from 0.0 to 1.0 that is created by summing the cube of the fraction formed by dividing the count in the color bin by the total pixel count. Since the fraction is guaranteed to be of value 1.0 or less, the cube of the fraction decreases the value. The smaller the fraction the more quickly the cube of the fraction approaches zero. This method penalizes colors that are perceived to be infrequent, decreasing their perceived contribution to the color count. Thus, the higher the resulting fraction after the summation the fewer the perceived colors. Taking the fraction of color to any power greater than 1.0 will exhibit similar behavior, however, the powers near 3.0 appear to produce the most satisfactory results.

A simple, two color example clarifies this. Assume two images both having only two colors. Let the first image A have an equal amount of each color, but let the second image B be 90% one color and 10% the other. Image A will have a color score of $(0.5^3 + 0.5^3) = 0.25$, but image B will have a score of
5 $(0.9^3 + 0.1^3) = 0.73$. Image B had a significantly higher fraction which reflects the perception that it is dominated by a smaller set of colors. This technique may be used with any number of colors. The threshold above which an image is considered to be of limited color is chosen empirically.

At Block 570, the scoring process determines the relative symmetry of
10 the image and depresses the previous score by an amount relative to the degree of symmetry.

In practical application, background images are found frequently on home computer systems. Many applications provide sample images for users to select as backgrounds in designing reports, web pages, or other digital
15 documents or imagery. A key feature shared by backgrounds and many 'canned' images is a strong symmetry along the horizontal and/or vertical mid-lines or either diagonal. Empirical evidence shows that pornographic images have a relatively strong symmetry around the vertical mid-line (left half vs. right half), but weak to no symmetry around the horizontal mid-line (top vs bottom) and the
20 diagonals. In practice, any metric can be used to score each of these regions then comparisons for similarity and dissimilarity between each of the halves can be made.

Training on target images determines a reliable threshold to differentiate between target and non-target. In the scoring process simple color classification
25 is used in the four regions of the image created by dividing along the horizontal and vertical mid-lines. The four regions can be combined to represent top vs bottom, right vs left, top left/bottom right vs top right/bottom left. Each pairing combines the scores and the absolute difference between the opposite pairing is the measure of difference. A value above the threshold determined by training
30 indicates a lack of symmetry and, thus, less evidence of the image being a background.

Next at Block 575, the scoring process analyzes the image to determine if the image contains or exhibits curvature attributes. Accordingly, in one embodiment, little or no curvature is an indication of non-target imagery.

Generally, images of people exhibit edges with curvature. Thus, an
 5 image not possessing strong curved edges is unlikely to include people and therefore not be pornographic in nature. Any image processing technique to detect curves may be used here. This scoring process uses a fast but approximate curvature detection technique described herein.

The core of the curvature detection technique is to eliminate edges
 10 related to straight lines. As such, the remaining edges then belong to boundaries that are curved. In one embodiment, a Huff transform could be used for straight line detection, however, an approximation technique may be used here as well. In the first phase, the technique makes two passes over the image keeping two sets of counts. The first set of counts is the number of adjacent
 15 edge pixels in the row. At a break in the row, the count is reset to zero and the scanning continues. For example, if * is an edge pixel and '0' is not and given the following image row data, the counts are the following:
 1,2,3,0,0,1,2,3,4,5,6,0,0,0,0,1,0,0,1,2,3. Thus, there is a string of 3, 6, 1, and 3
 edge pixels in this row. A sweep from right to left renumbers pixels with the
 20 highest numbered pixel to its right before a zero is encountered. The last row shows the final numbering.

Row Data:	* * * 0 0 * * * * * 0 0 0 0 * 0 0 * * *
Phase 1 counts:	1 2 3 0 0 1 2 3 4 5 6 0 0 0 0 1 0 0 1 2 3
Phase 2 counts:	3 3 3 0 0 6 6 6 6 6 6 0 0 0 0 1 0 0 3 3 3

A similar process is done along the vertical columns. A final sweep
 25 omits any pixel whose value exceeds a threshold value in either the horizontal or vertical direction. These pixels are part of relatively long vertical lines in one or the other directions. The remaining edge pixels are part of curved lines. Too few such pixels indicate no strong support for curved lines and lack of people,

and consequently target imagery. In one embodiment, the techniques errs to registering false positives in the presence of strong straight lines at significant angles to either the horizontal or vertical directions.

At Block 580, the scoring process updates the image data in the database, wherein the score of the image is assigned and recorded in the image database. In one embodiment, any data from the intermediate analysis steps could also be recorded to save time in the event of a second phase of scoring selected images is implemented.

After recording the score for the image in the image database, the scoring process branches to Block 505, to determine whether there are any current images that are to be analyzed and scored and to process any remaining images.

With reference to the scoring process embodied in Figure 5, the order of the filters and operations is important in some cases. Subsampling first is practical for fixing the worst case analysis speed. Color filtering and texture filtering can be done in either order, or possibly combined. Doing color filtering first has the advantage in that images with an initial low score, such that the likelihood of the image being a target image is very low, permits processing the image to stop at this point and return the score (ratio of kept to total pixels). This is a result of the fact that additional processing can only further decrease the score, not increase it. This short cut significantly reduces unnecessary processing time. In color filtering, compensating for a limited color range is typically done before counting the occurrence frequency of colors. Filtering flat color regions can be done before or after the color compensation step, in actual practice. The shrink phase of the acceptable color regions may be done first to encourage the joining of relatively close unacceptable pixels. True skin regions will survive the shrinking and will regain their full dimension after the subsequent grow phase. In texture analysis, the gradient is taken first then edges/zero crossings. The second phase classifies pixels as being either in a rough or smooth region. A grow phase of the rough pixels followed by a shrink phase forms the regions with sufficient support for

rough texture. Looking for curvature and symmetry help identify images either computer generated or of man-made structures. The next step is to determine region size and remove small regions to penalize fractured images. The last step is to compensate for colors that are over-represented. This step should generally be last since it is based on global data of the image and there is no longer a spatial correlation between the pixels and the color histogram.

Figure 6 illustrates an embodiment of a viewing process which may be implemented in accordance with one embodiment of the present invention.

The image viewer is an important component for manual review of the results achieved by the scoring process. The design of the viewer is such that a user can easily review thousands of images an hour and interact with the viewer via the mouse to select or otherwise manage the image set. In one embodiment, the images displayed within all viewers, except the viewer based on colors, are displayed sorted by decreasing score, so the most interesting images (i.e., those most likely to contain pornographic material) are highly likely to appear in the first few panels. Images sharing the same score are then sorted by decreasing number of pixels so the largest images are displayed first within the set of a given score. In one embodiment, the viewer which displays images having a limited color range sorts the images by decreasing order of the number of pixels. The heuristic discovered is that interesting pornographic images will have higher resolution and, thus, more pixels. The numerical score of an image may be used internally and converted to one of three categories for printing purposes.

As mentioned above, the images are sorted by the internal score assigned during the scoring phase. A higher score denotes more elements of a pornographic image. The images are sorted in descending order thus the images most likely to contain nudity or pornographic content appear first. The viewer displays these images in an on-screen panel which can be sized to show any number of images per panel that will fit on the screen. As a general rule, users prefer to see more thumbnails per panel. As such, in one embodiment, the viewer dynamically sizes itself to fit within the screen's viewable area to show a

maximum number of images. Likewise, users may resize the viewer window to display fewer images per panel. In one embodiment, the number of images displayed depends on the viewing area. In one embodiment, the number of images displayed in the viewing area ranges between 12 and 54 images per page. This presentation contrasts with other products that keep the number of thumbnails the constant, but resize them to fit the available area.

In one embodiment, the image viewer or viewing process has five classes into which images are partitioned. It is understood, that there are a number of viewer types selectable by the user. In one embodiment, the user may see all the images together in one viewer, or may opt to view disjoint subsets of the images by size or by the number of colors. Accordingly, one class is the ALL class which includes all images found. A second class is the Limited Color class that contains images that appear predominately monochromatic or near monochrome images that tend to score low due to lack of color. The remaining three classes divide the images by size: Large, Medium, and Small. LARGE, MEDIUM, and SMALL are relative values. Their absolute values may change with changes in common image format sizes.

The scoring process rates images for pornographic content, but the accuracy can be affected by the characteristic of the image. The five classes offer different and helpful views of the data. For example, images with a limited color range typically score low since the richness of color used in one filter is absent. The Limited Color view groups this type of image into its own group for quick review. As another example, small images are more difficult to score accurately than large images due to the necessary loss of detail. Separating the classes allows users to quickly review the (generally) more interesting class of images first, which is the Large class. This is important when time is of the essence such as a review of a computer system in the field. Each viewer class may be displayed in its own viewer but the functionality of the viewers is generally the same.

Accordingly, upon initialization of a viewer, start up code queries the system for its screen size then constructs a viewer scaled to fit the screen

dimensions. The images are in the database and are ranked according to some criteria. For instance, in the ALL, Large, Medium, and Small viewers, the images are sorted by decreasing score. In the Limited Color viewer the images are ordered by increasing color range. In one embodiment, assume N images
5 can fit in the scaled viewer window. In one embodiment, all viewers images sharing the same value are subsequently ranked by decreasing size. The first N images are displayed. The viewer window can be resized. The number of images to fit the new viewer window is recalculated. Finally, the window is automatically shrunk to remove any border too narrow to fit a full image.

10 In one embodiment, the user moves through the images displayed by the image viewer using five navigation buttons. The NEXT button shows the next set of images in rank. For example, if the viewer displays N images the viewer appears displaying images [1,N]. Selecting NEXT displays images [N+1, 2N]. Conversely, selecting PREV will display the previous panel of N images. The
15 last button displays the last panel and the first button displays the first panel. Two other buttons are available that move to the panel halfway between the current and the first (last) panel. The user may opt to select, e.g., left double click, on an image to create a new window with an enlarged view of the selected thumbnail.

20 The user can select, e.g., right click, to toggle a flag indicating that an image is selected/not selected. In this particular embodiment, the border around the selected image becomes red when selected and green when not selected. The selected images can then be saved during the reporting phase. Other selection options can be included in the future. One option is to select an image
25 for complete erasure or printing. Another option is to perform additional processing on the selected images such as generating a unique ID that depends on the image data. Such an ID would be useful in creating a database of known images to which future images can be compared. Another feature is a button that selects all images of a panel when clicked. The reload button performs a
30 new sort on the images and redisplay the current panel. Images that have been selected appear first in the new ranking.

Referring back to Figure 6, initially at Block 600, the data structures and graphical elements of the display are initialized. The start index pointer that indicates the active panel of images is set to zero. The image database is sorted by the proper criteria depending on the viewer type selected (ALL, Limited
5 Color, Large, Medium, Small). The base window and its sub-elements are displayed, but the panel of images are not yet loaded.

Next, at Block 605, the viewing process loads the current panel of images. A panel typically includes the set of images in the range between the start index and start index + (N-1), where N is the number of images that will fit
10 in the window area. Accordingly, the viewing process proceeds to the main loop to wait for user input.

At Block 610, the viewing process enters the Main Loop waiting for user input. Provided Reload, Prev, Next, First, Last, >>, or << are selected, then the viewing process proceeds to Block 615 Change Panel Action. Alternately,
15 provided the selection tool, e.g., mouse is right clicked once, double left clicked, or double right clicked, then the viewing process proceeds to Block 620 Select Image.

At Block 615 vectors to proper subroutines are invoked to process Reload, Prev, Next, First, Last, >>, or << panel change actions.

20 Next, at Block 625, the viewing process executes a Reload, wherein the images in the image database are sorted by score, and then size within the individual sets having the same score. After executing a Reload, the viewing process branches back to Block 605.

At Block 630, New Panel (Prev, Next, First, Last, >>, or <<):

25 Calculates the new start index relative to the action selected. For Prev, the start index = $\max(0, \text{start index} - N)$. For Next, start index = $\min(\text{start index of last panel}, \text{start index} + N)$. For First, start index = 0. For last, start index = start index of last panel. For >>, start index is the start index of the panel that is closest to $(\text{start index} + \text{last index})/2$. For <<, start index is the start index of the
30 panel that is closest to $(\text{start index})/2$. With the new start index calculated, the viewing process branches back to Block 605.

Returning back to Block 620, Select Image, one of three actions occurred that selects an image. Accordingly, the viewing process vectors to the proper action.

Block 635, a toggle delete on/off action, an image may be marked for deletion, such as by double right clicking on the image. In one embodiment, the background may be turned yellow if marked for deletion or the background may be either red or green (depending on selected state) if not marked for deletion. Generally, the file is not deleted until the user is queried during the save process. After this action, the viewing process branches back to Block 610.

Block 640, a toggle highlight on/off action, the state of the image may be toggled between being selected and not selected, such as by a mouse right click on the image. A selected image is highlighted or accented in some manner. In one embodiment, the border around a selected image turns red while an unselected image has a green border. After this action, the viewing process branches back to Block 610.

Block 645, a show image enlarged action, a window is launched that displays the image under the cursor as enlarged with additional file information in a text window below the image, such as by left double clicking a mouse. After this action, the viewing process branches back to Block 610.

In one embodiment, the speed of the viewer is almost entirely dictated by the IO speed to read the images. The delay on loading a panel can be noticeable. To speed the process, the viewing process can cache the N (e.g., N=1000) largest images found during scanning, regardless of score. Caching pre-loads the images that would cause the greatest delay in the viewer. In one embodiment, other heuristics might be used to cache not only the largest images, but also the images with the highest scores so the first and most important viewer panels are displayed as promptly as possible. Properly maintaining the list of highest scores generally requires a list of the top M scores sorted and updated incrementally, generally available algorithms exist that do this efficiently. As the user views these images, they have the option of selecting an image for later reporting. In one embodiment, the outline of that

image then becomes red letting the user know that it was successfully selected. Of course, tagging a selected image can also be done by adding any kind of special text to the thumbnail's associated text field. Users can also blow up an image from the thumbnail size to a larger size for closer inspection. In one
5 embodiment, the viewer will not only display the larger image but also will show the path, file name where that image can be found, time of creation, modification, and last access. Users then have the option of saving their results, as is discussed below.

Figure 7 illustrates an embodiment of an archive process which may be
10 implemented in accordance with one embodiment of the present invention. The archive process provides a user the ability to save the results of the image analysis performed above.

When a user selects a save option associated with images or attempts to exit the system, the user is given an opportunity to save the data gathered on the
15 run. The user enters an identification string and then selects a directory for the results to be stored. The first category of data is a text version of all images found that includes the full path and filename, size in bytes, the creation date, last access date, last modification date, and the scoring result. This is the primary information to be saved.

20 Selecting the save option pops up a file browser so the user can select where to create the subdirectory of results. In the spirit of not modifying the disk being scanned, the floppy drive or another drive can be selected on which to store the results. The archive process can also support writing reports via a modem or network to a remote machine. In one embodiment, one practical
25 application scenario is to connect to a police station system (i.e., supervisory authority) from a suspects machine and upload the reports directly to the police system via e-mail or a file transfer protocol, such as FTP. This enables the user to search for files on a computer, select those files he/she will report on, and write a report without writing anything on the drive being examined. This is an
30 important feature to users who are members of the law enforcement community and want to present these results as evidence in court.

Additionally, the user has options to report other information as well.

The first report option is a text format which displays information on the score assigned to it along with the creation date, last access date, and file location information.

5 The second report option is the creation of an HTML file which displays the selected images in their actual size with the file location and size information below it. Any HTML browser can be used to load the HTML file and print the contents. In a scenario to transfer the reports remotely as described above, groups of files will be combined into an archive file such as
10 the ZIP or any similar format. Thus multiple files will be transferred as one and compressed to save bandwidth requirements. An HTML report of the selected images can be recorded, which includes copying the selected images to a destination directory and creating the HTML page with links to the image copies. The user may then opt to save a list of images that could not scored and
15 a copy of the images themselves, again, in an HTML report. In one embodiment, this category is saved in two phases. The first phase creates a preview list in an HTML document that the user can review before deciding to save the files. The second phase asks the user to commit to saving the files. A list of movies that were found can also be saved. Likewise, the saving is
20 performed in two phases similar to saving the unscored images. In one embodiment, the user is then asked if they would like the application to launch an HTML browser for the preview. The user may skip any of the reporting categories, except the text document dump. In one embodiment, the user, during preview of unscored images and movies, may select a subset to be
25 copied, rather than the saving all of the image data.

Referring back to Figure 7, initially at Block 700, upon entry into the archive process a file browser is opened for the user to select a destination directory where the reports are to be written.

Next at Block 705, the user selects the destination or archive directory.
30 Accordingly, at Block 710, the user inputs the name or identifier for the

selected destination directory. In one embodiment, this window takes a string from the user to use as an identifier in the reports.

Next at Block 715, the results are saved in a Named Archive. In one embodiment, this is accomplished by creating a text file comprising the
5 information from all of the files found. Next, the user is queried whether they would like to save any selected images. Answering yes to the query will cause a report of the saved images to be generated. Additionally, copies of the files may be made in a format suitable for printing (e.g., JPEG). Next, the user is asked if the unscored images should be saved. Likewise, by responding yes, a
10 preview report is generated and a window asking if the user would like to preview this file via a browser. This gives the user an opportunity to decide if the unscored images should be saved. The user is given a prompt to confirm or cancel the saving of unscored images. An identical flow is available to the preview then save movie files that were found. The archive process exits at this
15 point.

Empirical evidence shows that given a general population of images typically found on a desktop or portable computer disk the most interesting pornographic images tend to be the larger images in the population. Thus, one embodiment of the present invention uses the number of pixels, a metric of
20 image size independent of compression technique, as a hint to the significance of an image. In sorting for viewing, the primary indicator of likelihood of pornographic content is score, but between images of the same score larger images are moved to the front of the grouping as they are more likely statistically to contain pornography. Therefore, in one embodiment, this
25 heuristic is relied on when displaying the images with limited color range. Since an image of limited color range (e.g., a black and white image) will typically score very low, score is an unreliable measure of content. Image size (pixel count) has proven to be a good secondary indicator.

Extensions

In one embodiment, the color population compensation phase may be extended to use color-correlated intensity population information. For each color, record a distribution, possibly compressed, of the frequency of intensities that occur in the training data. The last phase then would use the color's intensity distribution to determine the fraction of pixels allowed. Specifically, how often is the color available at each intensity level, or each increment of intensity, say at each interval 1/16th or 1/8th the full range.

Generally, local information cannot determine the size of the regions classified as target data, but target regions should be more than a handful of pixels. The following is a very fast method to size regions. This is an approach for fast merging of pixels into regions that hinges on a unique application of the well-known Union-FindSet algorithm that performs union by rank with path compression. The problem with merging is that it typically requires multiple passes to clean up changes made along the way. In the worst case the number of passes is on the order of the number of pixels (thousands), thus, is an $O(n^2)$. This method requires two passes and due to path compression is $O(n \lg^* n)$, where $\lg^* n$ is the inverse of the Ackerman function and in practical applications is less than 4.

One embodiment of the algorithm is as follows:

```

/* Make every pixel its own region initially */
For each pixel set UnionFindArray[row, col]->parent to self and count to 1;
/* Scan image and merge regions */
For (row=0; row<max_row; row++) {
25   For (col=0; col<max_col; col++) {
       If any neighboring pixel [rn, cn] to pixel [row, col] has
       RootParent[rn, cn] != RootParent[row,col] AND
       Pixel[rn, cn] is classified as a target pixel (i.e., not omitted) then
       /* Merge regions */
30   {

```

Update UnionFindArray as described in UnionFind

algorithm

specifically:

Set region 2's root parent to the root parent of region 1.

5 Add count of region 2 to count of region 1.

 }

 }

}

SetofRegions = Scan UnionFind structure and identify pixels whose root parent
10 is itself.

Sort the SetofRegions by size and select the root parent of the N largest regions.
Pass through image one last time omitting all pixels whose root parent was not
selected.

Done.

15 Initially, make every pixel its own region by giving a unique sequence
number to each and initializing the UnionFind structure to make the pixel its
own parent and have a count of 1. Now we merge regions as follows below.

Starting with the final mask created just before the color compensation
phase, make one pass over the mask row by row from left to right and
20 progressing in this manner top to bottom. The following steps generally only
apply to pixels labeled in the mask as target data.

For a given pixel, for each of its eight surrounding pixels, if the root
parents differ then merge the two regions in the following manner. (The root
parent is the pixel pointed to by following the parent pointers until one points to
25 itself. As one progresses up the chain of pointers the previous pointers are
updated to eventually point to the true root parent. Let region 1 be the region
having the root parent with the greater count. The root parent of region 2 is set
to the root parent of region 1. The count of region 1 is incremented by the count
of region 2.

30 At the end of the scan the Union-Find structure has a minimal set of
sequence numbers whose parent value is their own value. The number in the set

is the number of distinct regions and their sizes are given in the count fields. Sorting by size and taking the top N regions, set a flag for all sequence numbers not having as a parent one of the selected regions. In another pass over the mask, look up the flag for the pixel's sequence number and mask the pixel if it
5 does not belong to a region being kept. The final mask has small regions removed and has decreased the number of regions to a reasonable number. This depresses the ratings of highly fragmented images.

Given an image n -pixels this algorithm has two passes and run-time of $O(n \lg^* n)$ due to the properties of the UnionFind structure.

10

Ratio of Flesh Colors

One feature of pornographic images is the focus on male/female genitalia. These regions exhibit a noticeably different range of skin tones than non genitalia. By training on skin regions of the genitalia and non genitalia
15 separately two color maps will be generated. An additional filter to be added measures the area of non genitalia skin and the area of genitalia. An empirically determined threshold for the acceptable ratio of the areas of the two types of regions can be determined to differentiate between images displaying genitalia and those without.

20

Managing Limited Image Resources

In the field while taking evidence a system crash is catastrophic to the process. Often the source of a system crash on a personal computer is due to the system having insufficient resources for an application.

25

In one embodiment, in an effort to avoid such crashes, at start up the detection process scans the system for the amount of available memory to the application. Knowing the maximum memory available to it, the detection process adjusts its behavior to ensure safe operation within the system's resource memory constraints. The detection process first verifies that the
30 amount is above 8MB just for the core algorithm. If the system does not have that amount the application notifies the user and then ends. If there is sufficient

memory for the application core, the detection process then calculates the number of images that can be cached for viewing. The number of images to be cached satisfies the following conditions. The total memory less the 8MB for the core application, less a buffer amount of memory, e.g., 8MB, that will not be
5 used. The remaining memory is divided by the amount of memory required per viewer thumbnail, e.g., 48KB, to determine the number of images to that can be cached. If a certain minimum number of images cannot be cached then caching is not enabled since its effects would be negligible and disabling it lessens the chance that another application launched by the user could cause a system
10 cache. Of course, in sensitive environments the detection process strongly encourages the application run with as few other applications open as possible.

Decoding images to expand them to their full pixel representation requires space. In the 8MB reserved at start up by the detection process, 6MB are for image decoding. In one embodiment, image headers are read to
15 determine their pixel size before they are decoded. Images that require more than 6MB of memory space are not decoded or scored. These large images are listed as a separate category in the unscored images directory during the report phase.

In one embodiment, the detection process may dynamically adjust the
20 technique to match processor memory resources (score larger images), processor speed (more/less aggressive algorithm), and storage capability for reporting (style/detail of final reports).

Extremely Accurate Detection Using Grouping

25 An application of a well-known statistical technique that is very powerful and can be applied to the detection process for use as an extremely accurate web filter or to enhance the accuracy of in determining the content of any group of images. Given the image analysis core, a distribution of scores across a population of images can be estimated from sampling. A large set of
30 target images will produce a profile of the distribution of target scores. Similarly, a run on a set of non-target data will produce a similar distribution for

non-target scores. A basic theorem of statistics can then be applied to provide a filter with arbitrarily high accuracy.

In one embodiment, the statistical technique is implemented as follows.

The distributions are probability distributions on the likelihood a given score represents either a target image or a non-target image. With the core the accuracy of the distributions for any single image is about 95%, i.e., the image can be classified correctly 95% of the time. This means an error rate of 5% or 1 in 20. This error rate is high for large volume traffic.

Provided, on the other hand, the scores from a group of images were taken and averaged the probability distributions will become very nearly Gaussian and the variance will narrow greatly. This tightening of the spreads means that a group of non-target images will almost always have their average score very near the non-target distribution's mean and likewise for groups of target images. Significant deviation from the mean becomes rare unless the image types in the group are not what one assumed. Given this fact, one can test the result of scoring N images against the probability that the resulting averaged score is from a population of N non-target images. At a per image 95% accuracy rate, only a handful of images have to be target images to make the probability very low that the whole group is (incorrectly) classified as non-target. Here the definition of accuracy is changed to represent the probability that a set of images contains target data rather than the accuracy of an image in particular. The achievable level of accuracy for a set of images can be made arbitrarily high (> 99.99%) by having an increasing number of images in the set.

An application for the web is when a user clicks to a new website a filter can download not only the requested root page, but also the sub-pages at the same url before the page is displayed. As images are scored a table lookup determines the probability that the last M images are from a population of strictly non-target images. The fetch ahead scoring stops when enough images have been fetched to achieve the desired level of confidence in the results (this is known directly from the distributions). With the accuracy of such a technique (95%) this may only require five to ten images, which might be one to three

pages at a typical pornographic site. At this point the entire site is either accepted or rejected (easily determined by the url pathname). Thus, any penalty is paid only once upon entering the site. A list of acceptable and unacceptable sites can be cached so revisits do not incur needless delays. At less than one
5 second per image, the filter could score an image while retrieving the next so the delay is simply that to download the images.

Interestingly, this technique is orthogonal to image analysis. The text-based web filters currently being sold can easily incorporate this statistical technique into their tools. Another application is in the disk scanning process
10 where directories of images are ranked so only significant caches of images are flagged. Any grouping of images can be tested in this manner (series of e-mail attachments to a user, web site, and disk directories) to improve the accuracy of detecting pornographic images. This saves time by allowing system administrators to focus solely on the big offenders.

15 The core image analysis technique can be dynamically adjusted to a higher level of sensitivity and computation intensity for groupings that indicate a high probability of containing pornography. The advantage of such a dynamic capability is to increase accuracy with only a marginal increase in processing time in typical environments where the number of non-pornographic images
20 greatly outnumbers pornographic images.

Attaching Technology to a Web Spider

As hinted at above, a spider can roam the web and apply the classification algorithm to individual images, web pages, or groups of web
25 pages found on the web. This application makes the spider a proactive use of the technology which contrasts sharply to the reactive use such as would be found in web filtering that requires a user to initiate the image or web page transfer. Conversely, such a spider can be used to automatically gather pornography as well.

30

Detecting Pornography in Movies

There are two basic environments: off-line and on-line analysis. Off-line analysis means the entire movie is available for analysis before a classification determination is required. For example, law enforcement agencies would like to automatically analyze MPEG or QuickTime digital movies to determine if they are pornographic in nature. They want a summary of the information so they do not have to waste time viewing the entire movie. On-line analysis is the process of filtering a continuous video feed, possibly with some lead time, to discover pornographic frames while being viewed.

Off-line analysis has the advantage that all the data is available to the analysis tool. This allows the system to sample frames so only a fraction have to be analyzed (e.g., one of out thirty frames, or once per second of film). Some research has been done in determining scene change points, so classification can be made on a scene-by-scene basis. Since many images are available, increasing accuracy via groups of images applies here. However, since within a scene the images are not statistically independent, different distributions are required to provide accurate confidence intervals. The distribution would be formed by scenes that appear before or after a scene. Groups of scenes that score as having pornography will support the hypothesis of nearby scenes that also score as having pornography. Conversely, scenes that score as not having pornography will decrease confidence in nearby scenes that score as having pornography. Using nearby scenes as additional evidence to support/refute the content of a given scene will increase accuracy.

Figure 8 illustrates one embodiment of a typical movie scene to which one embodiment of the present pornography technique may be applied. Distance becomes a factor that decreases the further away two scenes are separated. A standard mathematical representation often used of such a factor is of the form Ae^{-Bd} , where A and B are constants and d is the measure of distance, alternately, a Gaussian distribution centered over the scene of interest may be used. Applications would typically be for law enforcement or customs agents that have a video (either tape or MPEG) that is digitized, analyzed, and then

displayed. The display can either be a set of sample images statically displayed scoring the highest or a reorganization of the scenes (digitally) so a viewer can quickly view the movie "chopped up" but with the area desired to be viewed first. A blend of the two techniques would have a static thumbnail of a scene
5 that if clicked plays the whole scene for the viewer. This allows very quick human review of the entire movie and de-emphasizes the need for accurate rating.

On-line analysis takes a "live" feed and must detect target content immediately, but the analysis is essentially the same, however, only previous
10 scenes can be used for supporting hypothesis. Applications for this would be a TV filter or censor for homes with children. The image stream has to be digitized for analysis, thus, analog TV signals have to be converted but high definition TV signals may be able to be processed directly. The challenges for on-line analysis are in the need for very high accuracy. People are unlikely to
15 tolerate frequent errors that mask off 'clean' video or do not mask 'dirty' video. Unfortunately, a 95% accuracy rate is unlikely to be sufficient. A technique for tolerating this uncertainty is to provide various levels of intervention that range from subtle to obvious as confidence in the scoring goes from low to high, respectively. Solutions include fuzzifying out detected regions or blanking the
20 screen. Blanking the screen is intolerable if an error in judgment is made, however, making the screen or regions of it fuzzy may be more tolerable to people since the show is still somewhat viewable in the case of errors and the overall story line can be followed. A hardware filtering box could buffer frames to delay the display until after analysis. This provides the analysis tool a
25 window of frames to generate a greater confidence in the analysis results. All images will be of consistent quality which allow tuning the filters to a much tighter tolerance. High definition TV will provide excellent data and a special hardware filter can run very fast analysis, possibly at frame rate of 1 image per 1/30th of a second. Another feature to help users tolerate errors is a hot button
30 using a key on the remote so parents can force the filter ON or OFF during

viewing. The aggressiveness of the filtering can also be user programmable ranging from erring to less filtering to erring to more aggressive filtering.

General Markets/Applications

5 The ability to select and completely delete target images from within the system: Let user ID an image via a mouse click, create a list of selected images in memory, and upon exit query the user with at least a dialog box to confirm the delete and a viewer displaying all the images to be deleted. The user can then click to undelete. The delete is not actually performed until the tool is
10 exited. This allows the user ample leeway to rethink a delete. Upon deletion the system ensures the image is absolutely unrecoverable. This can be done by overwriting the portion of disk with the image. The system will conform to all accepted standards/methods for making data irretrievable.

Incremental Runs: Run only on files that were changed since last run.
15 Keep a log of file names, creation times, modification times, and possibly a checksum so any changes will be detected.

After scanning, the found images are compared to the database (stored on disk) from the last scan. Only files created or modified after then are scored. Or, files with different checksums indicating a surreptitious change.

20 Exclusions: Select an image which won't be searched again unless the checksum and/or the modified date changes. This is a selective subset of the Incremental Runs method. An additional flag is stored in the log. Run against an input file of image file pathnames.

Instead of scanning a disk, skip this step and use a file of pathnames
25 input by the user instead. This is a swap of the scan module with a file read module.

Run from a network: There are two ways this can be done. First, have the remote disk mounted on the machine running the detection process. All accesses look to be local, but are actually going across the network. The second
30 method is to have a client-server architecture where each machine to be scanned has a server daemon running. The client runs on a remote host where a user can

graphically select the machine to be scanned. The client then sends a message to the server daemon on the selected machine which kicks off the scan.

Upon completion, the server can either ship a copy of the image files to the client machine or, if the server is a webserver, return an html file with
5 hyperlinks to the images. This technology can be bifurcated along a path of related products involving the searching and displaying or blocking of pornographic images and other generic image analysis products.

A business model that is possible is to provide a run-once version that a user downloads from the web for a nominal fee. Upon getting the download
10 request the detection process server compiles a version of the software with an expiration date in the near future after which the software will no longer run.

Other pornographic focused related technologies are: Software V-Chip for Digital TV. Tune the filter to the high quality images of digital TV.

The method of filtering / detection requires routing the signal through a
15 digital processing unit that runs the algorithm. Regions of suspect content can be blacked out, fuzzied, or the whole picture could be blacked out. The hardware box that does the processing can also have an override to force filtering on/off via an unused button on the remote control. The capability to force filtering on/off can be disabled/enabled via a password key sequence from
20 the remote. This will allow parents to compensate filtering decisions they disagree with.

The detection process can be used to scan ISP servers or corporate proxy servers and/or firewalls to ensure they do not have child pornography on their systems. Corporations can use this technology to monitor their employees who
25 request pornographic images.

The detection process as an Internet image based filtering/blocking product (consumer market). Just as the smut filters stream through the text before a page is displayed on a browser, the detection process filters screens images. In Netscape™, such filters are called plug-ins. The filter can be
30 combined with the text-based approaches. The detection process is a product that scans email attachments or embedded images and blocks them if deemed

pornographic (or that detects a hotlink in an email, searches the site automatically and deletes email if deemed pornographic). This entails 'decoding' a file to identify it as a document of a type that might contain an image, scanning the file for images, and pulling out the image data for analysis.

- 5 Document types include e-mail attachments, word processing documents, postscript files, and any other type of document that permits embedding an image. Taking the face shot of a kiddie pornographic image and matching it up with databases to determine the identity of that child. This task is the much studied face recognition problem. The difference in the present invention is the
- 10 user interface that lets the human select the interesting area to clip out before the matching algorithms are applied and the final use.

The technology can be applied to any device which is capable of displaying or transmitting or receiving an image file in an electronic format.

- The following is a non-exclusive list of areas to which the core
- 15 technology can be applied:
- E-mail servers to identify e-mail attachments
 - Proxy server to filter web traffic
 - Internet devices- Computer, PDA, cellphone, images downloaded from web
urls.
 - 20 ▪ Digital TV streams to the receiver
 - Video Servers to filter the broadcast stream
 - Internet meeting applications
 - Products to search for pornography on computer media or the web
 - DTV Return channel filtering
 - 25 ▪ DTV Tunneled Data filtering
 - Internet cache filtering
 - Home Networking systems
 - MPEG streams to a set top box, or set top storage device
 - ATSC DASE compliant PC Cards
 - 30 ▪ Video Phones
 - Electronic Books

- Suppliers to Digital Broadcast, Cable, DBS, or Broadband
- Pornography Filter for VCRs, Camcorders, Digital Cameras, DVD players and recorders, etc.
- Internet Picture Frames that download images that get displayed in an electronic picture frame
- Internet photo albums.

The result of the filtering can be proactive or as simple as setting flag that defers to the device to implement a preferred filtering protocol that can range from doing nothing to complete blockage of the offending material.

Possible actions include, but are not limited to:

- Do nothing
- Record incident in a log for later reference
- Display a warning but leave content viewable
- Blur the regions of the image that are rated as questionable
- Block image altogether
- Determine actions on a scene by scene basis
- Send an email to a designated person with information on the policy violation.

Broadcast Market

Figure 9 illustrates one embodiment of Broadcast Market capable of implementing the teaching of one embodiment of the present invention.

Opportunities for insertion of detection process technology into the broadcast and consumer electronics chain:

At the broadcast channel: The broadcast channel loosely refers to any entity engaged in dissemination of content over public or private broadcast channels. Content to be broadcast is supplied from the broadcast signal source, which could include a television network, cable network, satellite network, broadband network, private business network, etc. The local broadcaster or cable head end refers to the local or regional broadcast distribution point for the

television, cable, satellite, or broadband network, which may or may not supply local/regional content in the broadcast stream. The broadcast signal source and/or the local broadcaster or cable head end may choose to include content from the internet which they may or may not have authored. Interactive service providers and interactive interface equipment manage the interactions generated by users at their receivers, delivered through the interaction network.

The detection process technology can be implemented at any point in the broadcast channel where content from the internet could be inserted.

Specifically, these points would be the broadcast signal source, the local broadcaster or cable headend, or at the interactive service provider/interactive interface equipment. Typically at the broadcast channel, this content would be managed via a content server, so the detection process technology would reside as a software tool which may be used to do real time scans of various file types as they're being served into the broadcast stream, or periodically scheduled scans of file types. The technology would by no means be limited to use in this application. The detection process technology could also be inserted into various manufacturers specialized hardware, either in the drivers or in the hardware design, used in the overall systems configurations of these broadcast signal sources, local distribution points, and interactive service providers.

At the receiver: The receiver is loosely defined as any device that can receive broadcast media. Specific examples of receivers would include television sets, set top boxes (STB's), PC/TV receiver cards, digital set-top recording devices referred to as personal video recorders (PVR's), video cassette recorders (VCR's), software enabling receiver capabilities on a PC-like device, PCS phones, handheld devices, and all "converged" multi-function devices which enable receipt of broadcast media. The detection process technology could be used either as a hardware implementation or software implementation in the broadcast interface, interactive interface, and/or media and application processing components of the receiver.

In addition to these applications, the detection process technology could also be implemented in home networking applications. Due to the development

of several competing home networking standards, consumers will be able to network all consumer devices in their homes which conform to the specific standards. This means that at any point in the home networking chain where media can be stored, transmitted, or viewed, the detection process technology
5 could be implemented in hardware or software to filter images.

Special Discussion on a Software V-Chip Application

In contrast to the hardware V-Chip technology currently required in all TVs sold in the United States, the detection process can be used in a software
10 application in digital TVs to provide additional functionality that the hardware V-Chip cannot.

The hardware V-Chip reads simple signals in the VBI (vertical blanking interval) voluntarily provided by the broadcasters that provide information on the rating of a show. In the absence of such information the hardware V-Chip
15 can provide no protection other than blanking unknown programs. A software V-Chip including the core image analysis algorithm can provide on the fly detection of pornography and selectively blank or fuzz out offending scenes.

A contributing idea is the definition of per scene ratings. This idea is compatible with both hardware and software V-Chips. The idea is to make
20 ratings fine grained to a scene level. The advantage is in movies that may have only a few scenes a parent may find objectionable are now predominately viewable since only those scenes viewed objectionable are omitted.

Furthermore, the logical extension is to allow encodings that permit directors to provide editing directions in the data stream that project alternate scenes and or
25 dialog depending on the filtering level currently enabled.

Method:

1. Codec unit takes in MPEG encoded signal.
2. Decoded frames are sent to the analyzer and a frame buffer.
3. The analyzer scores the frame and correlates the current frame with previous
30 frames and scenes.

3b. The analyzer may optionally do scene change detection and use this information in a dynamic fashion to determine which frames to analyze.

4. If target imagery is detected a flag is sent to the frame buffer, possibly with bounding box information delimiting the region of offense.

- 5 5. Frame buffer takes flag and bounding box information and takes the appropriate actions as dictated by the user selected filtering policy. Actions include blocking the signal, fuzzing out the offending region, logging the event, e-mailing a user specified account. If no flag is sent then the frame is sent unchanged.

10 Analysis could be on every frame or a sampling of the frames. Figure 10 illustrates one embodiment of a system capable of implementing the teachings of the present invention within a V-Chip environment.

Correspondence Filter: Detect links and keywords in Chat windows; fires off a report to systems administrator or parents. Keyword detection is
15 what the text-based filters do. Links can be tested via text- and image-filtering to detect questionable topics being discussed in a cloaked fashion. Parents might like to have such a tool to help monitor the chat rooms their kids are on. The output can be silently stored for later review by the parents.

Other generic image analysis products are: Detection of counterfeit
20 currency. The detection process does this in a simple fashion by color analysis primarily since any scanned images of near-life-like money are illegal. Exact template matching can be developed for very strict matching criteria can extend the application to detect counterfeit money on the fly, i.e., at point of sale: stores, vending machines, etc. A hardware device takes in a bill, shines a bright
25 light on it and through it, digitize the image(s) then applies color analysis, OCR (optical character recognition), tests for reflecting strips and microtext, and model matching to known images. A database of known frequent counterfeit cues are kept and very targeted tests developed and applied. A list of known counterfeit serial numbers is easy to keep and is the easiest data to automatically
30 read off the bills using OCR. U.S. Secret Service Application is obvious. The same is obvious for the law enforcement agencies in other countries responsible

for preventing counterfeiting of their currency. The same application will be valuable to banks. This also can function as an add-on device to Point of Sale systems .

The detection process is also used for the detection of unauthorized use
5 of logos (attach to spider and search www). Similar to counterfeit detection, but less stringent. Given the logo to match, it is broken into its primary components of color, shape, texture, etc. via known techniques and test images taken from web pages for a close match. A second phase applied to those passing the first filter makes a much more constrained test of matching to the original image.
10 Techniques include methods to use invariant features described earlier in this patent that compensate for slight shifts in color, size, shape (distortion horizontally or vertically). Every image scanned can be kept in a database tagged with the web page it was from so new requests can be compared to previous searches. The image matching can also be done for exact matches to a
15 give logo. This method would fingerprint the image by generating a unique code from the image data. Well known checksum techniques are available such circular redundancy codes (CRC). While very fast, this technique is easier to evade by making imperceptible shifts in the image.

Advertisement verification system. Using the method just described for
20 logos a system can randomly sample a web site and record the ad images. The logo matching system detects the number of times a particular add appears. Using statistical sampling techniques a confidence level can be developed to verify that the terms of an advertising contract are being fulfilled in that a company's ads are being displayed as frequently as was agreed. Again storing
25 the images in a database allows the system to review the history of accesses for other companies. Both techniques described above for image matching can be use, either exact matches or use of invariant features.

Figure 11 illustrates an embodiment of a machine-readable medium
1100 containing various sets of instructions, code sequences, configuration
30 information, and other data used by a computer or other machine processing device. The embodiment of the machine-readable medium 1100, illustrated in

Figure 11, is suitable for use with the image analysis process described above.

The various information stored on medium 1100 is used to perform various data processing operations. Machine-readable medium 1100 is also referred to as a computer-readable or processor-readable medium. Machine-readable medium
5 1100 can be any type of magnetic, optical, or electrical storage medium including a diskette, magnetic tape, CD-ROM, memory device, or other storage medium or carrier-wave signal.

Machine-readable medium 1100 includes interface code 1105 that controls the flow of information between various devices or components
10 associated with the image analysis process. Interface code 1105 may control the transfer of information within a device, or between an input/output port and a storage device. Additionally, interface code 1105 may control the transfer of information from one device to another.

In addition, machine-readable medium 1100 also includes scanning
15 code 1110, scoring code 1115, viewing code 1120, and archiving code 1125, which are configured to perform the respective operations discussed above, namely, the scanning process, scoring process, viewing process, and archiving process, respectively.

From the above description and drawings, it will be understood by those
20 of ordinary skill in the art that the particular embodiments shown and described are for purposes of illustration only and are not intended to limit the scope of the invention. Those of ordinary skill in the art will recognize that the invention may be embodied in other specific forms without departing from its spirit or essential characteristics. References to details of particular embodiments are
25 not intended to limit the scope of the claims.

CLAIMS

What is claimed is:

- 5 1. A method of detecting target data in image data, the method comprising:
scanning elements contained in a data structure for image data;
scoring the scanned image data, wherein a score value associated with
the scanned image data indicates the likelihood of target data in the scanned
image data; and
10 viewing the scored image data for the target data.
2. The method of claim 1, wherein the target data comprises pornographic
content.
- 15 3. The method of claim 1, further including the step of:
archiving the scored image data containing the target data.
4. The method of claim 1, wherein the data structure is specified by a user.
- 20 5. The method of claim 1, wherein the scanning of elements contained in the
data structure is accomplished using system calls that update access dates
associated with the image data.
6. The method of claim 1, wherein the scanning of elements contained in the
25 data structure employs a depth first search to traverse the data structure.
7. The method of claim 1, wherein wherein the scanning of elements contained
in the data structure:
traversing the data structure;
30 reading data segments from the elements contained in data structure; and

comparing signature values associated with the data segments to the position of the data segments within the elements to determine whether the element is image data.

5 8. The method of claim 7, further including the step of:

classifying the element as image data provided the signature values correspond to the proper position of the data segment within the element.

9. The method of claim 1, wherein scoring the scanned image data comprises:

10 sampling a subset of pixels contained in the image data;

applying a color compensation scheme to the sampled image data;

omitting pixels in the sampled image data which are surrounded by identically colored pixels creating a color mask;

mapping pixels in the sampled image data to an accepted color set;

15 performing a texture analysis, provided the sampled image is greater than a specified image size, wherein select pixels not representing target textures are omitted creating a texture mask;

merging the color mask and texture mask to create a final mask; and

20 generating an initial score from the ratio of pixels not omitted in the final mask.

10. The method of claim 9, wherein applying a color compensation scheme to the sampled image data comprises:

25 smoothing the sampled image data by averaging values surrounding each pixel.

11. The method of claim 9, wherein image regions under a specified size are removed after merging the color mask and the texture mask.

30 12. The method of claim 9, further including the steps of:

performing color compensation that depresses the contribution of pixels whose color is over represented;

calculating a color count factor representing the perceived range of colors in the image;

5 determining a relative symmetry of the image and depressing the initial score by an amount relative to the degree of relative symmetry;

examining the image for curvature attributes; and

assigning a score to the original image data.

10 13. The method of claim 1, wherein viewing the scored image comprises:

loading a current panel of images;

inputting user image selection action to invoke corresponding viewer action;

sorting images by score for presentation to user; and

15 calculating a new start index relative to the selection action.

14. The method of claim 1, wherein viewing the scored image comprises:

selecting an image; and

marking an image for deletion.

20

15. The method of claim 1, wherein viewing the scored image comprises:

selecting an image; and

enlarging the image with additional image file information.

25 16. The method of claim 3, wherein archiving the scored image data comprises:

selecting a destination directory;

inputting an identifier for the selected destination directory; and

saving results of image analysis.

30 17. A method of detecting and ranking target data in image data, the method comprising:

sampling a subset of pixels contained in the image data;
applying a color compensation scheme to the sampled image data;
omitting pixels in the sampled image data which are surrounded by
identically colored pixels creating a color mask;
5 mapping pixels in the sampled image data to an accepted color set;
performing a texture analysis, provided the sampled image is greater
than a specified image size, wherein select pixels not representing target
textures are omitted creating a texture mask;
merging the color mask and texture mask to create a final mask; and
10 generating an initial score from the ratio of pixels not omitted in the
final mask.

18. The method of claim 17, wherein applying a color compensation scheme to
the sampled image data comprises:

15 smoothing the sampled image data by averaging values surrounding
each pixel.

19. The method of claim 17, wherein image regions under a specified size are
removed after merging the color mask and the texture mask.

20

20. The method of claim 17, further including the steps of:

performing color compensation that depresses the contribution of pixels
whose color is over represented;

25 calculating a color count factor representing the perceived range of
colors in the image;

determining a relative symmetry of the image and depressing the initial
score by an amount relative to the degree of relative symmetry;

examining the image for curvature attributes; and

assigning a score to the original image data.

30

21. A machine readable medium containing executable instructions which, when executed in a processing system, causes the processing system to perform the steps of a method for detecting target data in image data, the method comprising:
- scanning elements contained in a data structure for image data;
 - 5 scoring the scanned image data, wherein a score value associated with the scanned image data indicates the likelihood of target data in the scanned image data; and
 - viewing the scored image data for the target data.
- 10 22. The medium of claim 21, wherein the target data comprises pornographic content.
23. The medium of claim 21, further including the step of:
- archiving the scored image data containing the target data.
- 15 24. The medium of claim 21, wherein the data structure is specified by a user.
25. The medium of claim 21, wherein the scanning of elements contained in the data structure is accomplished using system calls that update access dates
- 20 associated with the image data.
26. The medium of claim 21, wherein the scanning of elements contained in the data structure employs a depth first search to traverse the data structure.
- 25 27. The medium of claim 21, wherein the scanning of elements contained in the data structure comprises:
- traversing the data structure;
 - reading data segments from the elements contained in data structure; and
 - comparing signature values associated with the data segments to the
 - 30 position of the data segments within the elements to determine whether the element is image data.

28. The medium of claim 27, further including the step of:

classifying the element as image data provided the signature values correspond to the proper position of the data segment within the element.

5

29. The medium of claim 21, wherein scoring the scanned image data comprises:

sampling a subset of pixels contained in the image data;

applying a color compensation scheme to the sampled image data;

10

omitting pixels in the sampled image data which are surrounded by identically colored pixels creating a color mask;

mapping pixels in the sampled image data to an accepted color set;

performing a texture analysis, provided the sampled image is greater than a specified image size, wherein select pixels not representing target

15

textures are omitted creating a texture mask;

merging the color mask and texture mask to create a final mask; and

generating an initial score from the ratio of pixels not omitted in the final mask.

20

30. The medium of claim 29, wherein applying a color compensation scheme to the sampled image data comprises:

smoothing the sampled image data by averaging values surrounding each pixel.

25

31. The medium of claim 29, wherein image regions under a specified size are removed after merging the color mask and the texture mask.

32. The medium of claim 29, further including the steps of:

performing color compensation that depresses the contribution of pixels

30

whose color is over represented;

calculating a color count factor representing the perceived range of
colors in the image;
determining a relative symmetry of the image and depressing the initial
score by an amount relative to the degree of relative symmetry;
5 examining the image for curvature attributes; and
assigning a score to the original image data.

33. The medium of claim 21, wherein viewing the scored image comprises:
loading a current panel of images;
10 inputting user image selection action to invoke corresponding viewer
action;
sorting images by score for presentation to user; and
calculating a new start index relative to the selection action.

15 34. The medium of claim 21, wherein viewing the scored image comprises:
selecting an image; and
marking an image for deletion.

35. The medium of claim 21, wherein viewing the scored image comprises:
20 selecting an image; and
enlarging the image with additional image file information.

36. The medium of claim 23, wherein archiving the scored image data
comprises:
25 selecting a destination directory;
inputting an identifier for the selected destination directory; and
saving results of image analysis.

37. A machine readable medium containing executable instructions which, when
30 executed in a processing system, causes the processing system to perform the steps

of a method for detecting and ranking target data in image data, the method comprising:

- sampling a subset of pixels contained in the image data;
 - applying a color compensation scheme to the sampled image data;
 - 5 omitting pixels in the sampled image data which are surrounded by identically colored pixels creating a color mask;
 - mapping pixels in the sampled image data to an accepted color set;
 - performing a texture analysis, provided the sampled image is greater than a specified image size, wherein select pixels not representing target
 - 10 textures are omitted creating a texture mask;
 - merging the color mask and texture mask to create a final mask; and
 - generating an initial score from the ratio of pixels not omitted in the final mask.
- 15 38. The medium of claim 37, wherein applying a color compensation scheme to the sampled image data comprises:
- smoothing the sampled image data by averaging values surrounding each pixel.
- 20 39. The medium of claim 37, wherein image regions under a specified size are removed after merging the color mask and the texture mask.
40. The medium of claim 37, further including the steps of:
- performing color compensation that depresses the contribution of pixels
 - 25 whose color is over represented;
 - calculating a color count factor representing the perceived range of colors in the image;
 - determining a relative symmetry of the image and depressing the initial score by an amount relative to the degree of relative symmetry;
 - 30 examining the image for curvature attributes; and
 - assigning a score to the original image data.

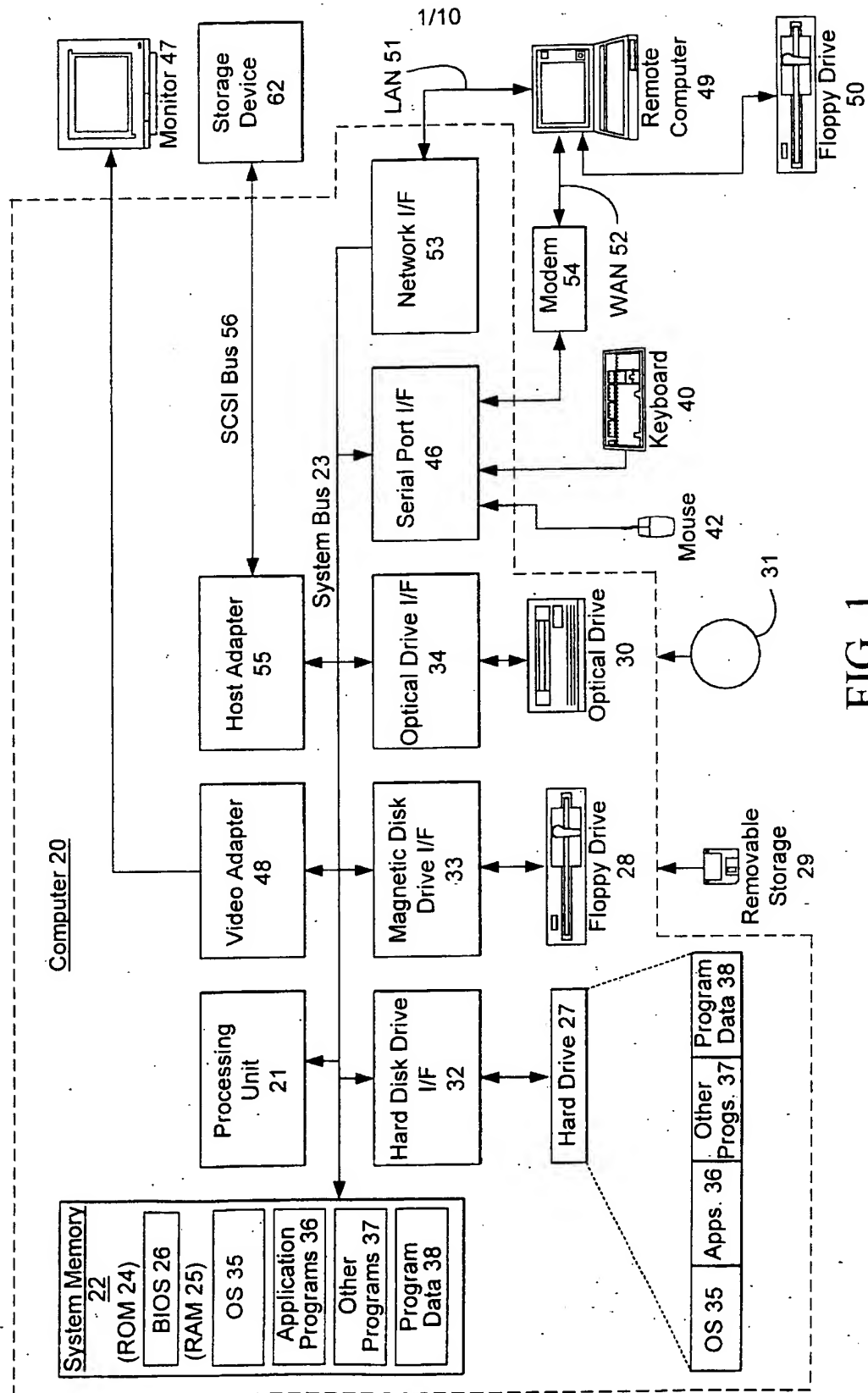


FIG. 1

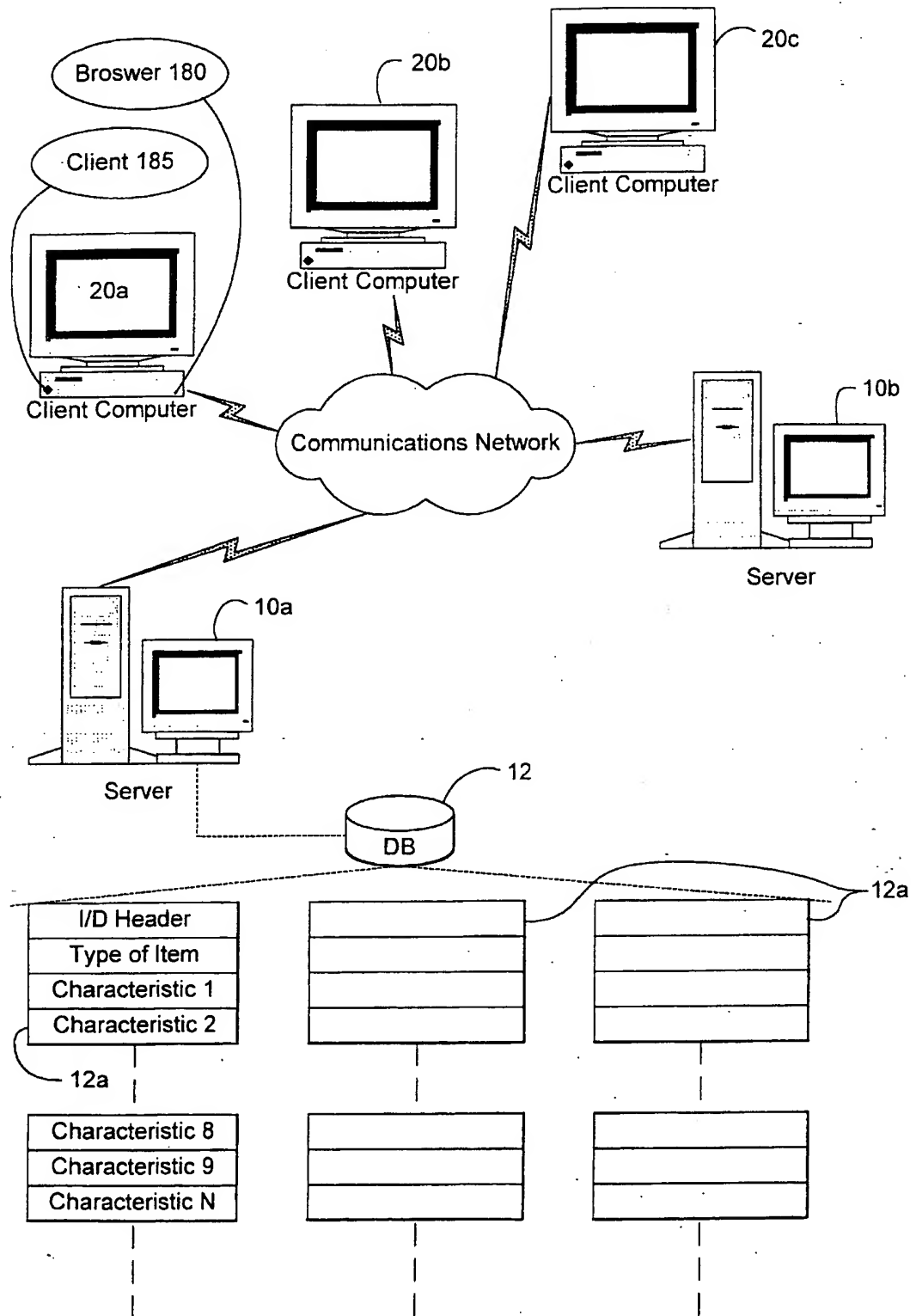


FIG. 2
SUBSTITUTE SHEET (RULE26)

3/10

Pictuate Program Processes

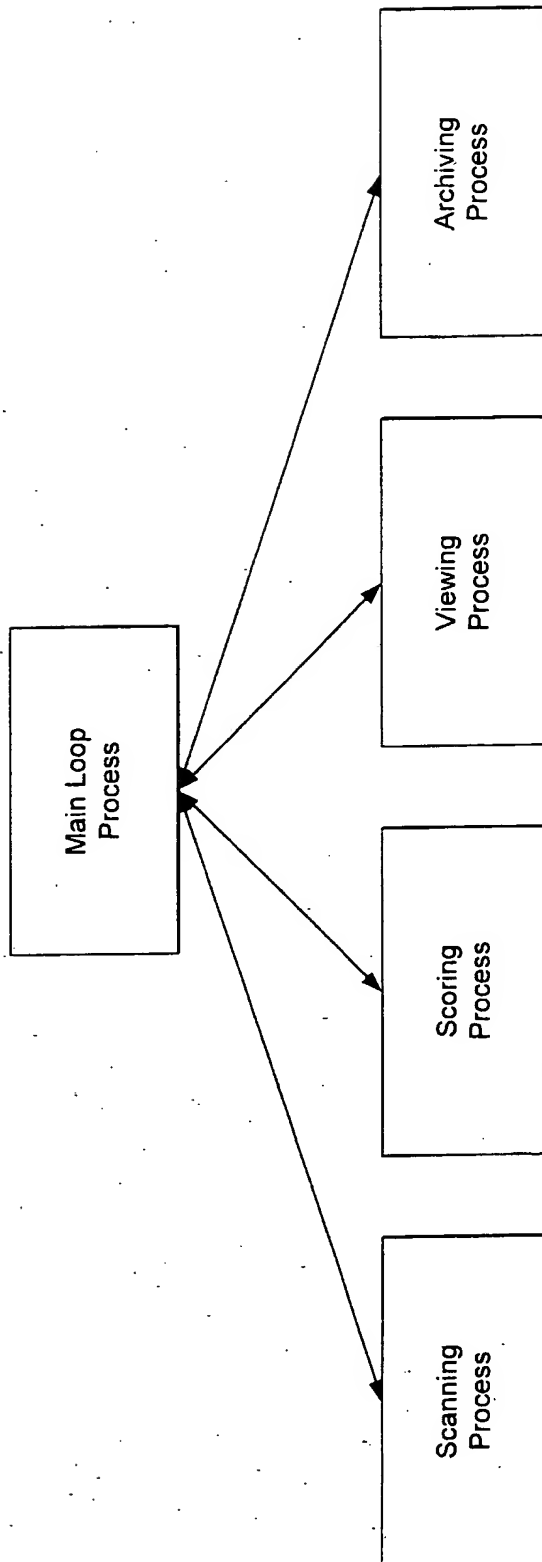


FIG. 3

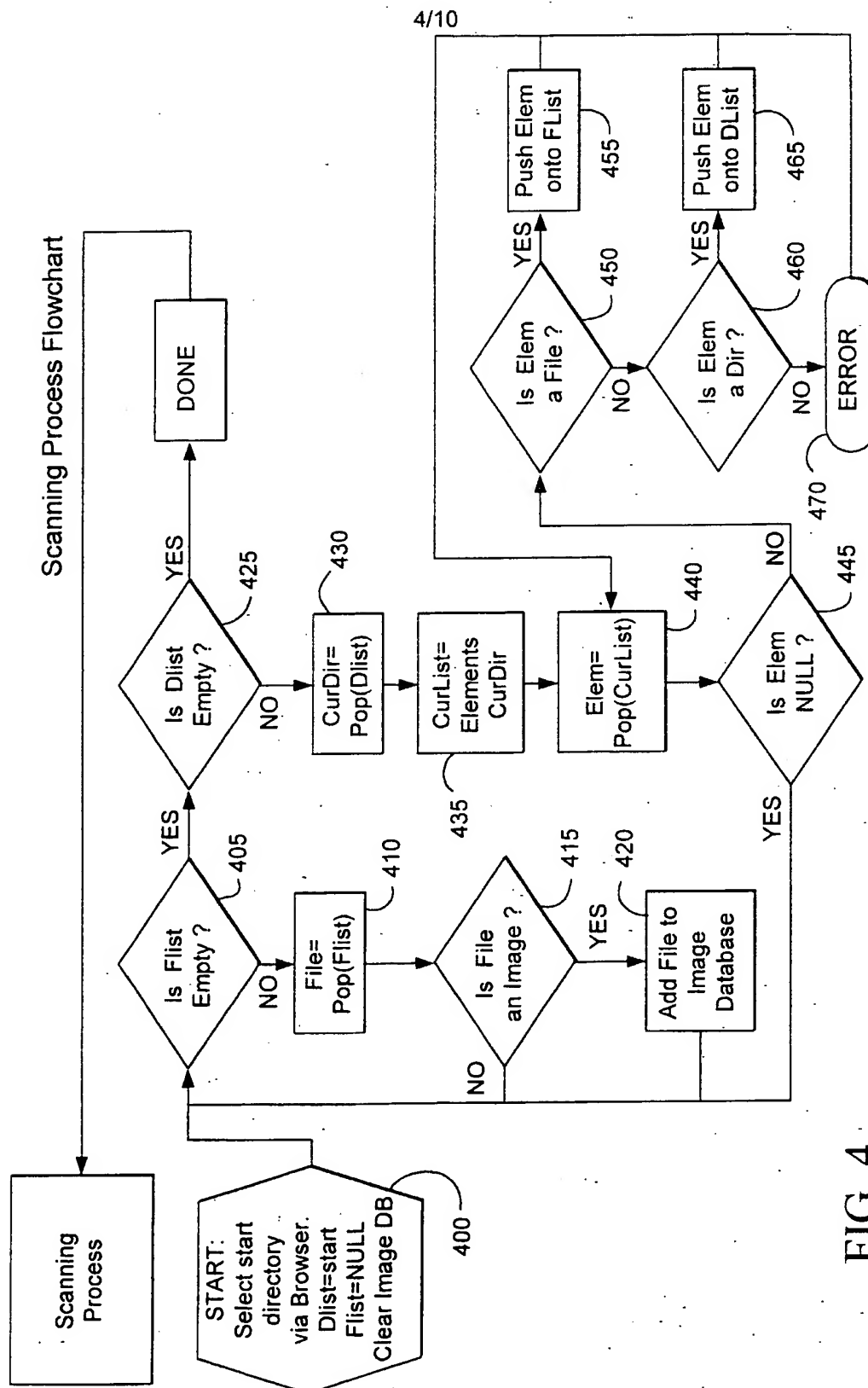


FIG. 4

5/10

Scoring Process Flowchart

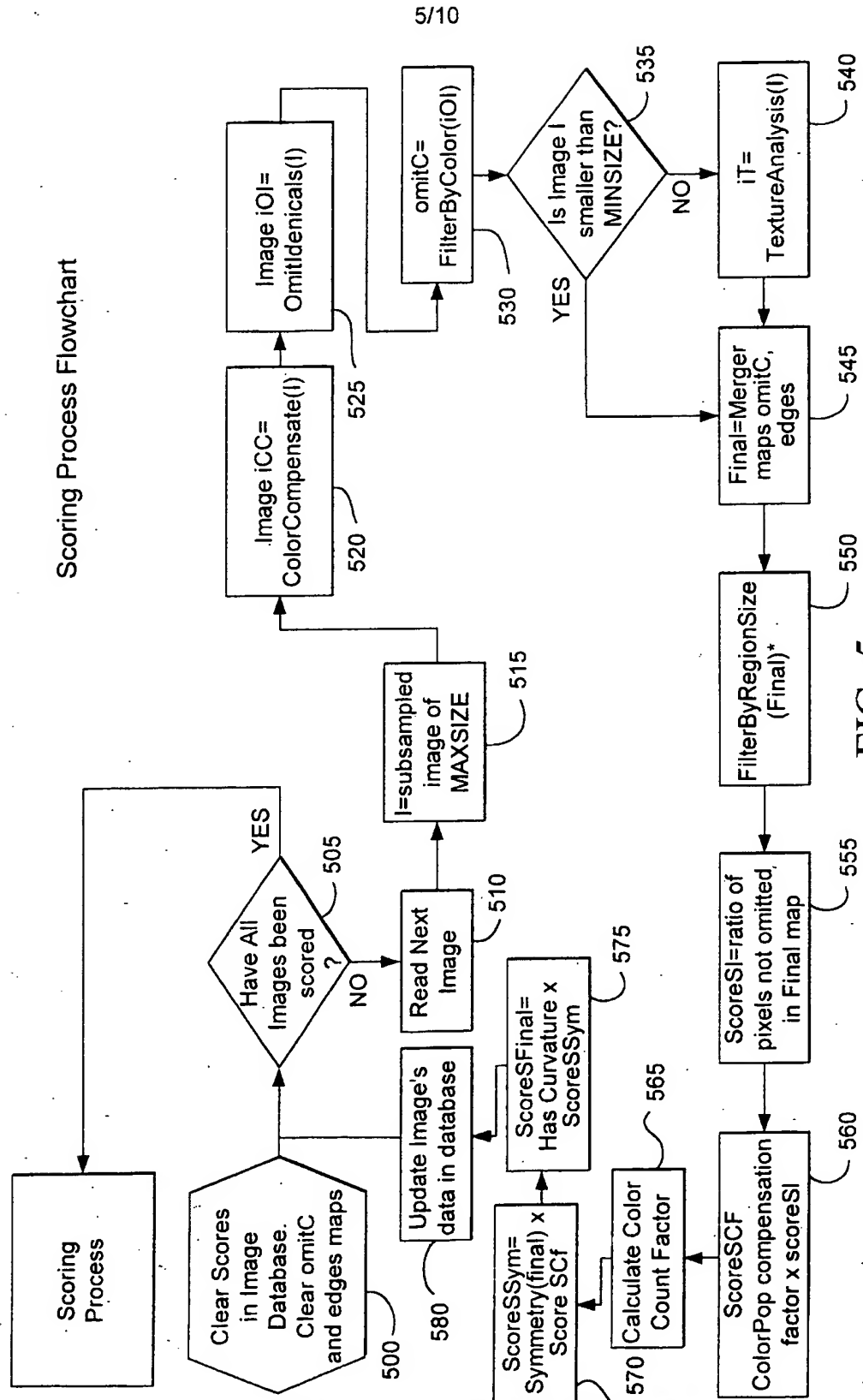


FIG. 5

6/10

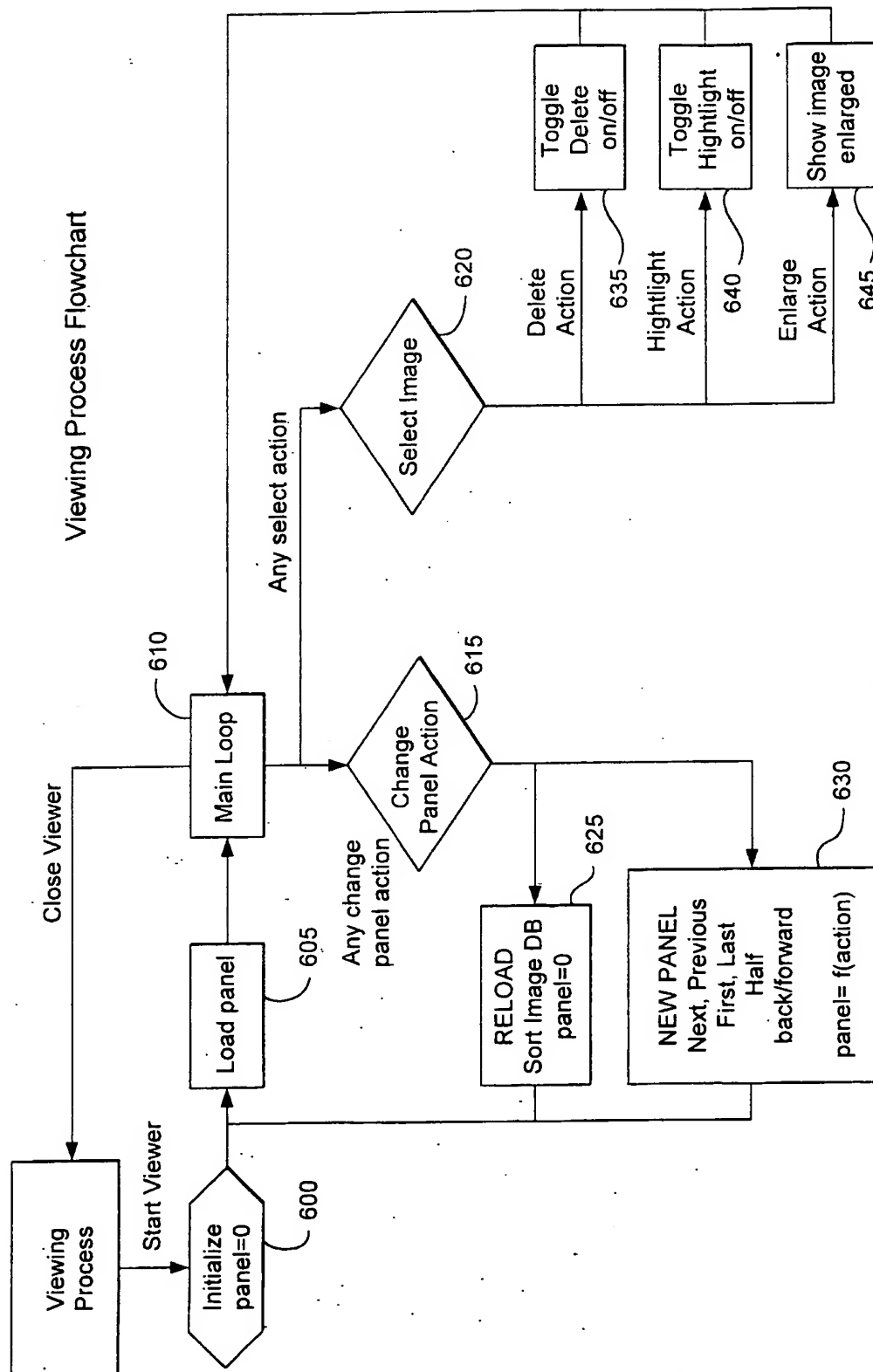


FIG. 6

7/10

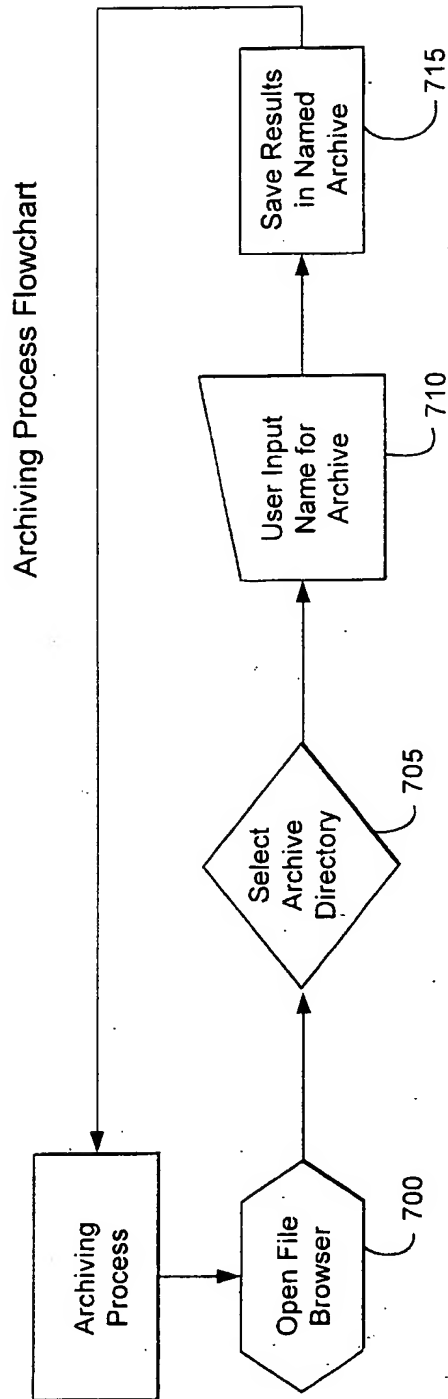


FIG. 7

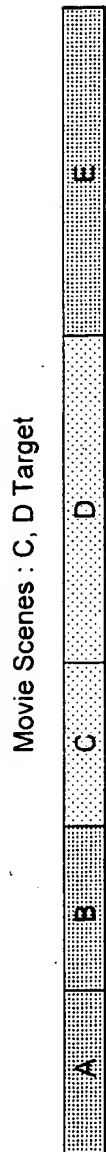


FIG. 8

These elements can be co-located or remote

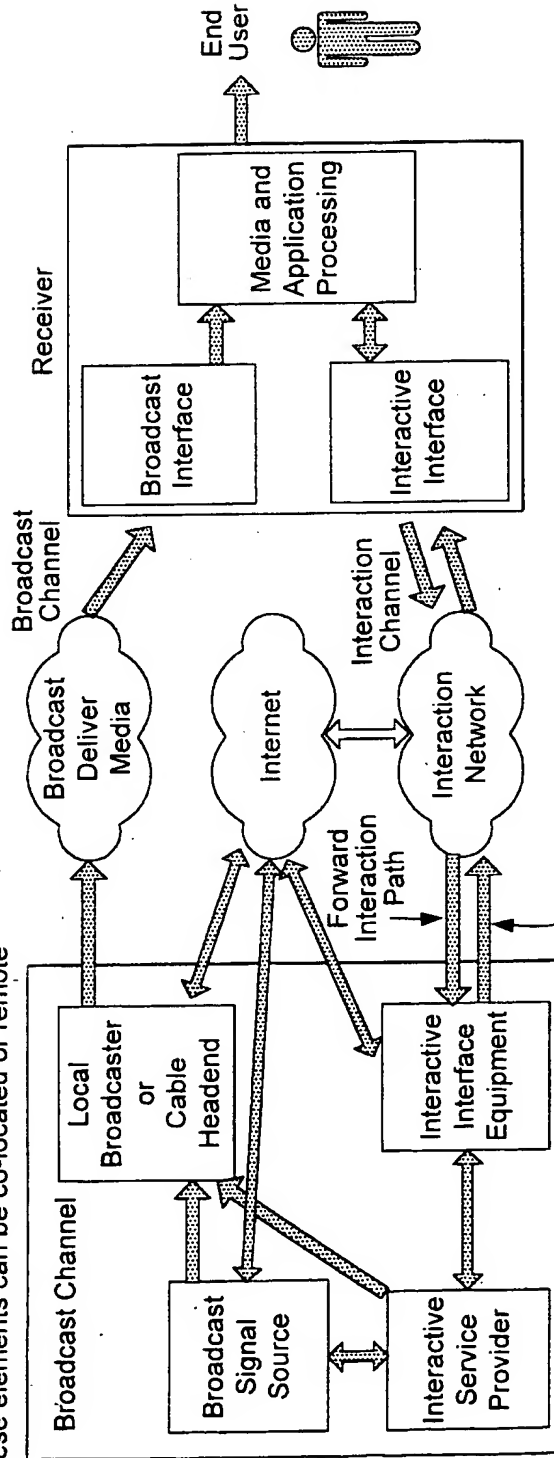


FIG. 9

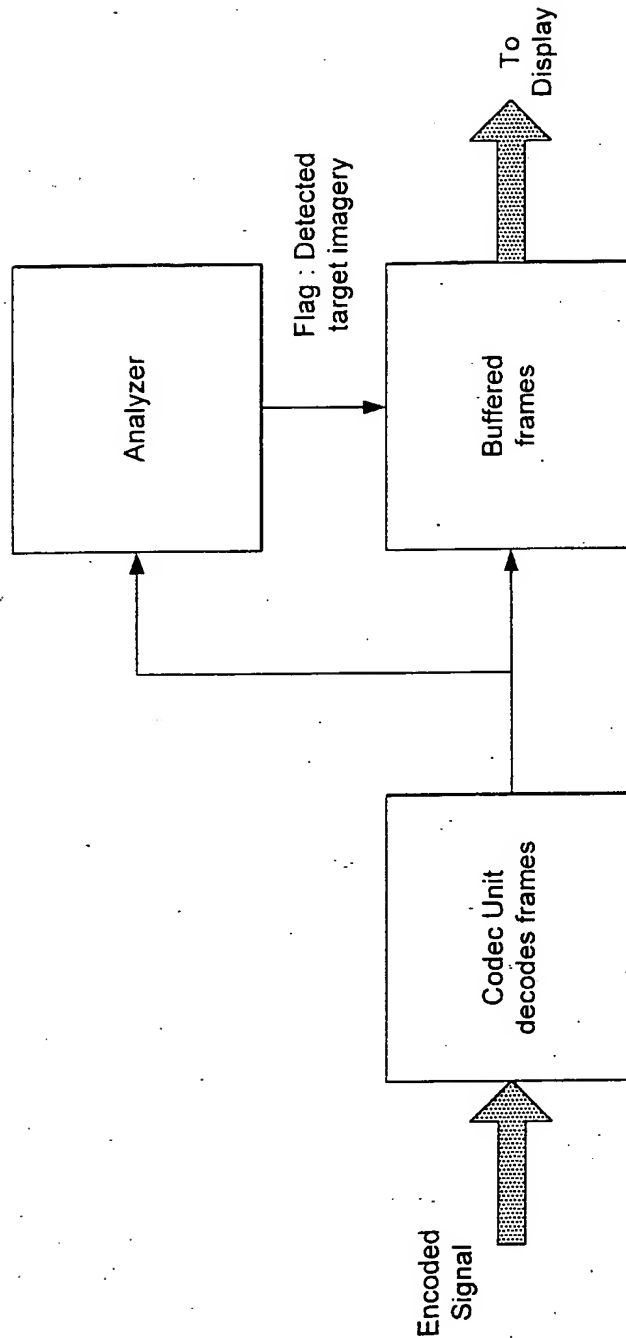


FIG. 10

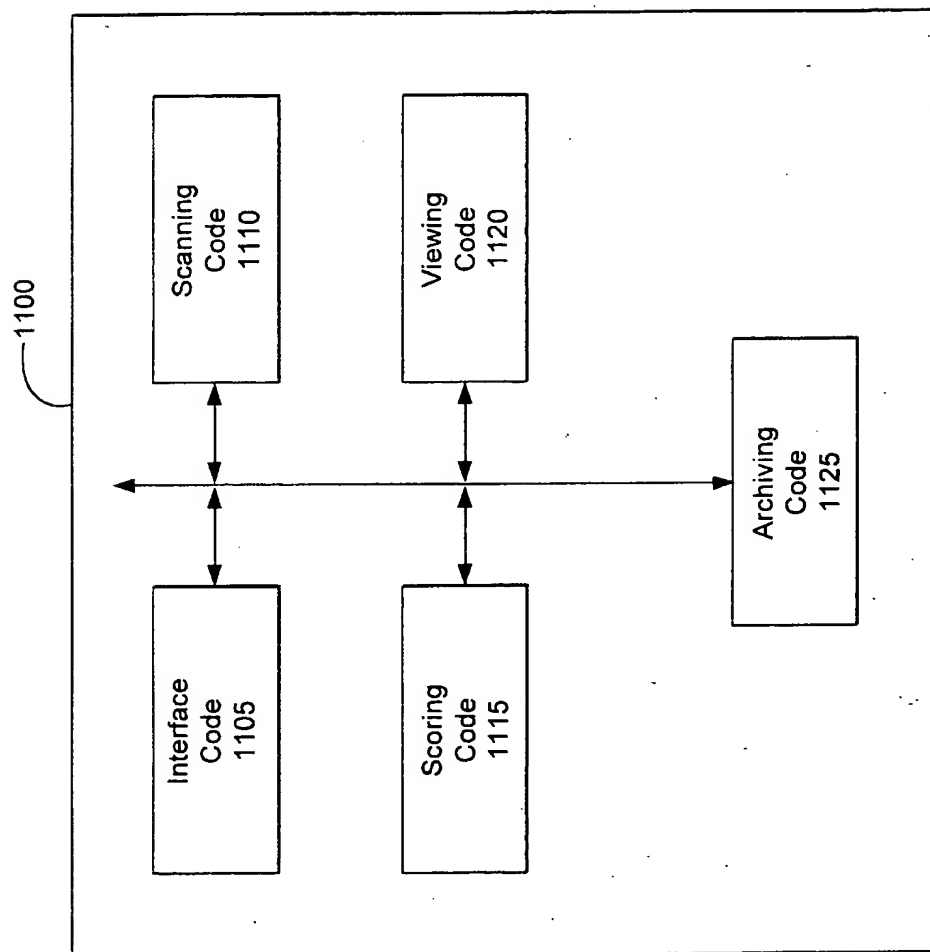


FIG. 11

This Page Blank (uspto)